NUWC-NPT Technical Document 10,560
15 February 1996

Application of the Finite Element/Boundary Element Approach to the Analysis of Radiation and Scattering from Fluid-Loaded Elastic and Piezoelectric Structures

Michele D. McCollum
Robert E. Montgomery
Clementina M. Siders
Underwater Sound Reference Department

Naval Undersea Warfare Center Division
Newport, Rhode Island

Approved for public release; distribution is unlimited.

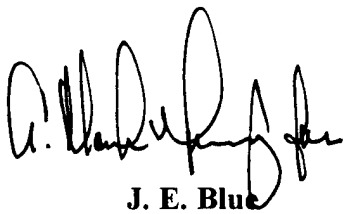
19960812 149

PREFACE

This document was prepared under Project No. S72571, principal investigator M. D. McCollum (Code 2571). The sponsoring activity is the Office of Naval Research.

The technical reviewer for this document was A. L. Van Buren (Code 258).

Reviewed and Approved: 15 February 1996



J. E. Blue
Head, Underwater Sound Reference Department

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

Public reporting for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington Headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188), Washington, DC 20503.

1. AGENCY USE ONLY *(Leave blank)*	2. REPORT DATE	3. REPORT TYPE AND DATES COVERED
	15 February 1996	Final

4. TITLE AND SUBTITLE

Application of the Finite Element/Boundary Element Approach to the Analysis of Radiation and Scattering from Fluid-Loaded Elastic and Piezoelectric Structures

5. FUNDING NUMBERS

PE - 0601153N
TA - R5391
WU - DN580-030

6. AUTHOR(S)

Michele D. McCollum, Robert E. Montgomery, and Clementina M. Siders

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)

Naval Undersea Warfare Center
Underwater Sound Reference Detachment
P. O. Box 568337
Orlando, FL 32856-8337

8. PERFORMING ORGANIZATION REPORT NUMBER

TD 10,560

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)

Office of Naval Research
800 N. Quincy Street
Arlington, VA 22217-5000

10. SPONSORING/MONITORING AGENCY REPORT NUMBER

11. SUPPLEMENTARY NOTES

Work was performed while the Underwater Sound Reference Detachment was part of the Naval Research Laboratory, Washington, DC.

12a. DISTRIBUTION/AVAILABILITY STATEMENT

Approved for public release; distribution is unlimited.

12b. DISTRIBUTION CODE

13. ABSTRACT *(Maximum 200 words)*

To understand the behavior of realistic fluid-loaded elastic or piezoelectric structures, it is often necessary to use numerical techniques. The approach chosen by many Navy modelers has been to combine a finite element model of the structure with a combined Helmholtz integral equation formulation (CHIEF) boundary element model of the wetted surface. CHIEF computes a matrix that describes the interactions through the fluid of the elements on the wetted surface. The CHIEF matrix can be combined with the structural matrices from any finite element program to obtain the fluid-loaded response, in the frequency domain, of a device or structure. This approach has been used to compute the harmonic response to mechanical force, an applied potential, and an incident acoustic plane wave. The most recent enhancement to the finite element/CHIEF combination is the estimation of the in-fluid eigenfrequency of a specific mode of the structure. Some of these procedures have been documented in various Navy reports, while the most recent applications have been described in journal articles. The purpose of this document is to present all the procedures in one publication, and explain and demonstrate their use for the benefit of the prospective modeler or designer.

14. SUBJECT TERMS

Acoustics
Acoustic Scattering
Finite Element Analysis
Boundary Element Analysis
Modal Analysis
Combined Helmholtz Integral Equation Formulation

15. NUMBER OF PAGES
138

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
Unclassified	Unclassified	Unclassified	SAR

TABLE OF CONTENTS

TABLE OF CONTENTS (Cont'd)

LIST OF ILLUSTRATIONS

Figure **Page**

LIST OF TABLES

Table **Page**

LIST OF ABBREVIATIONS, ACRONYMS, AND SYMBOLS

ATILA	*Analyse du Tranducteurs par Integration des Equations de Laplace* (finite element code)
BE	Boundary element
CHIEF	Combined Helmholtz integral equation formulation (boundary element code)
DOF	Degree of freedom
E	Elastic
FE	Finite element
H	Harmonic
I	Current
IE	Infinite element
M	Modal
OC	Open circuit
P	Piezoelectric
S	Surface area
SC	Short circuit
Y	Input electrical admittance
Y	Young's modulus
c	Speed of sound
d	Diameter
f	Cyclic frequency
j	$\sqrt{-1}$
k	Electromechanical coupling constant
$\bar{k}$	Wavenumber
ℓ	Length
ndof	Number of degrees of freedom
nsurf	Number of boundary elements
$\{p\}$	Surface pressure
$\{q\}$	Nodal charge
ν	Poisson's ratio
$\{v_N\}$	Normal velocity distribution
η	Loss factor
$\{\phi\}$	Electrical potential
λ	Acoustic wavelength
ρ	Density
Ω	Radian excitation frequency
ω	Radian frequency
ω_i	Set of radian frequencies

LIST OF ABBREVIATIONS, ACRONYMS, AND SYMBOLS (Cont'd)

Matrices

$[A]$, $[B]$	Boundary element matrices
$[C]$	Compatibility matrix
$[K]$	Stiffness matrix
$[M]$	Mass matrix
$[R]$	Re$\{[Z]\}$
$[X]$	Im$\{[Z]\}$
$[Z]$	Mutual coupling matrix
$[d]$	Matrix of piezoelectric strain constants
$[s^E]$	Matrix of compliance constants
$[\varepsilon^s]$	Matrix of dielectric constants

Vectors

$\{\mathbf{f}\}$	Vector of applied nodal force
$\{\mathbf{u}\}$	Vector of nodal displacement
$\{\psi_i\}$	Set of eigenvectors

APPLICATION OF THE FINITE ELEMENT/BOUNDARY ELEMENT APPROACH TO THE ANALYSIS OF RADIATION AND SCATTERING FROM FLUID-LOADED ELASTIC AND PIEZOELECTRIC STRUCTURES

1. INTRODUCTION

There are a variety of methods available for modeling the linear, frequency-domain behavior of fluid-loaded structures, including analytic models, finite elements,[1] boundary elements,[2] and infinite elements.[3] Analytic models generally involve a closed-form solution to a set of approximate equations. For simple designs, analytic methods are useful because solutions for various geometric and material parameters can be obtained rapidly. For most realistic designs, however, analytic methods do not describe the level of detail necessary for accurate predictions of the in-fluid behavior. In these cases, an accurate description requires the use of numerical models based on finite elements (FEs), boundary elements (BEs), and infinite elements (IEs). It is assumed, for the purposes of this document, that the structure is modeled using finite elements, because this is the most common approach. An interior fluid domain can be modeled using either finite elements or boundary elements, while an exterior fluid can be described using any of the three types of elements.

Interior fluid domains, such as the air inside of an automobile, are commonly modeled with finite elements. Because the fluid volume is finite, the number of elements required is usually within reason. Also, with finite elements, visualization of the pressure field in the cavity is a straightforward problem. Boundary elements can also be used for this type of problem, with the advantage that they result in no additional degrees of freedom in the system of equations. However, it is not as easy to obtain a graphical representation of the pressure field because the user must specify the coordinates of each field point of interest.

Exterior fluid domains have also traditionally been described using finite elements. In this approach, one uses either damping elements or an absorbing boundary condition to satisfy the requirement that all energy travel away from, not toward, the radiating body. For two-dimensional models, the use of finite elements to model the exterior fluid domain does not add an inordinate number of degrees of freedom to the system. In this case, the main disadvantage of these elements is the need for specifying the extent of the finite element mesh. For three-dimensional models, however, the number of additional degrees of freedom associated with the fluid nodes can be computationally prohibitive.

Infinite elements offer a reasonable alternative to finite elements for modeling an infinite fluid domain. In this approach, a thin layer of finite elements surrounds the structural model, and a single layer of infinite elements surround the fluid finite elements. Unfortunately, these elements are not yet widely used and are available in a limited number of numerical modeling codes.

Another approach to modeling an infinite fluid domain is to use fluid boundary elements to describe the wetted surface of the structure. These are two-dimensional elements that are based on the Helmholtz integral equation.[2,4] Boundary elements offer much of the efficiency of infinite elements, and they are widely available. In fact, the boundary element code CHIEF (combined Helmholtz integral equation formulation),[5] and its preprocessor XCID (X-Windows CHIEF Interactive Driver),[6] are available to government contractors for a nominal processing fee.[7] The various numerical techniques available for modeling the various aspects of the fluid-loaded transducer problem are summarized in figure 1.

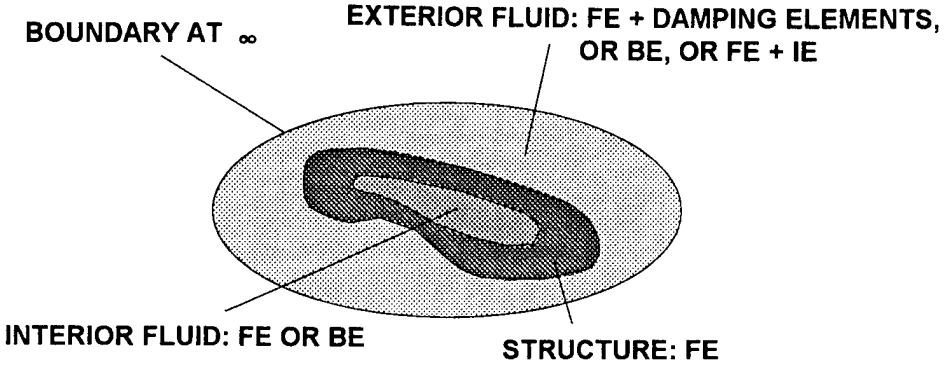


Figure 1. Methods for Modeling Fluid-Loaded Structures

Even though finite elements and boundary elements have been in use for decades, the combined FE/BE method is not widely used because of the difficulty of coupling the two techniques. Few companies offer fully integrated software packages using finite element and boundary element methods, and these packages are not affordable for many small transducer laboratories. The purpose of the present publication is to explain and demonstrate the procedure of coupling a finite element model with a boundary element model so that any laboratories that possess the two separate capabilities will be able to combine them in an external solving program. Alternatively, the user may use the equations to implement the procedure inside the finite element code, if the source code is available, or this can be done by the vendor of the code.

Section 2 presents a brief summary of the equations for the harmonic (forced) and modal (free) analysis of an *in vacuo* structure using finite elements. Section 3 describes the use of boundary elements for modeling an interior or exterior fluid domain. Section 4 combines the two methods in two coupled matrix equations, one for in-fluid harmonic analysis and the other for in-fluid modal analysis. This treatment of the coupled solution is intended to provide at least a basic understanding of the procedure and its components. Section 5 summarizes the equations and procedures, and section 6 describes possible pitfalls one might encounter in implementing the coupling procedure. The final section applies the coupled procedure to the analysis of radiation

and scattering from a simple fluid-loaded piezoelectric transducer, including the intermediate matrices and final results.

This document also includes four appendices. Appendix A provides a more theoretical development of the coupled equations. Appendix B gives a sample set of material constants describing a piezoelectric ceramic. Appendices C and D provide listings of files and programs related to the example problem, including a Fortran program for solving the *in vacuo* and in-fluid equations.

Before launching into an explanation of the FE/BE coupling process, it is important to specify the prerequisites. To be a general procedure, the coupling problem is solved in a computer program external to the finite element and boundary element programs. This requires that certain components of the solution be available for input to the external solver. Specifically, in terms of the finite element program, it is necessary to have a file containing the elastic, piezoelectric, and dielectric stiffness matrices, the mass matrix, and a matrix containing the nodal forces corresponding to an applied pressure on the wetted surfaces of the structure (all of these will be described in detail later in this document). In this way, any elastic/piezoelectric finite element code can be used, as long as it is capable of writing the matrices to an external file. If a particular code does not include this capability, it is possible to request a modification to the code. Otherwise, it is necessary to use a different code.

From the boundary element program, the mutual coupling matrix for both radiation and scattering analyses is needed, and for the scattering problem, the centroidal force vector is required. It is assumed in this document that a "patch" boundary element code is used. That is, the surface pressures and normal velocities are computed at the centroid of each surface element, and these quantities are assumed to be constant over the element. The boundary element code CHIEF is a patch code. A nodal boundary element code can also be used with only slight modification to the coupling procedure.[5]

2. FINITE ELEMENT MODEL

BACKGROUND

In a finite element model, a structure is discretized into a continuous set of elements and a corresponding set of nodes, which together describe the three-dimensional shape of the structure. The elements themselves may be two- or three-dimensional, depending on the assumptions that are made about the behavior of the structure. A typical three-dimensional element is a 20-noded brick,

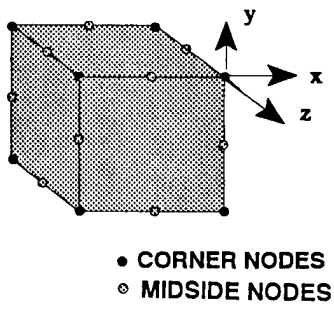


Figure 2. 20-Noded Brick Finite Element

shown in figure 2. This element has 8 corner nodes and 12 mid-side nodes, each of which may translate in the x-, y-, and z-directions. That is, each node has three translational, and no rotational, degrees of freedom (DOF). Suppose that this element represents a real structure, namely, a solid block , and that the elastic wavelengths in the material are much greater than the length of a side. Then the finite element model consists of just one element and 20 nodes. If the block is free to move in any direction, then the model has 60 degrees of freedom (20 nodes x 3 DOF/node = 60 total DOF). However, if we know that the bottom of the block is rigidly bonded to a rigid support structure, then the eight nodes on the bottom surface are fixed, so the model will have only 36 degrees of freedom [60 - (8x3) = 36].

So far, only the geometry of the structure has been described. Next the properties of the material from which the physical block is constructed must be specified. If the material is isotropic, such as steel, then one needs to specify the Young's modulus Y, Poisson's ratio ν, and possibly, the loss factor η. Once these quantities are specified, one can use the finite element code to compute the elastic stiffness matrix and the consistent mass matrix of the model. These matrices have dimensions (ndof x ndof), where ndof is the number of DOF in the model, in this case, 36. Each row and column pertains to a particular degree of freedom of the system. The finite element code assigns the order of the degrees of freedom, usually based on the criterion that the resulting matrices be as banded as possible. For example, DOF #3 might be the x displacement at node 6. The particular order does not matter, so long as the correspondence between nodal displacements and degree of freedom indices is known. This correspondence is referred to here as a degree of freedom table.

Once the stiffness and mass matrices have been computed, they can be used to determine the eigenvalues and eigenvectors of the system (modal analysis), as well as the response of the system to a specified excitation at a given frequency (harmonic analysis).

IN VACUO HARMONIC ANALYSIS

The equation for a harmonic analysis with a known forcing function is as follows:

$$\{[K] - \Omega^2 [M]\} \{\mathbf{u}\} = \{\mathbf{f}\} , \qquad (1)$$

where Ω is the radian excitation frequency, $[K]$ is the stiffness matrix, $[M]$ is the mass matrix, $\{\mathbf{u}\}$ is the vector of nodal displacements, and $\{\mathbf{f}\}$ is the vector of applied nodal forces. The term $\{[K] - \Omega^2[M]\}$ is called the dynamic stiffness matrix. Note that equation (1) does not include the effects of fluid loading.

To specify displacements, rather than forces, equation (1) takes a slightly different form. In this case, one first reorders and divides, or partitions, the rows and columns of the stiffness and mass matrices according to unknown and known displacements, as follows:

$$[K] = \begin{bmatrix} K_{RR} & K_{RA} \\ K_{RA}^{t} & K_{AA} \end{bmatrix} \tag{2}$$

and

$$[M] = \begin{bmatrix} M_{RR} & M_{RA} \\ M_{RA}^{t} & M_{AA} \end{bmatrix}, \tag{3}$$

where the subscript R indicates the reduced matrix, corresponding to unknown displacement degrees of freedom; the subscript A indicates the applied or known displacement degrees of freedom; and the superscript t represents the transpose of the partition. The columns corresponding to known displacements are now moved to the right-hand side of the equation, and the associated rows are eliminated, leaving the reduced stiffness and mass matrices $[K_{RR}]$ and $[M_{RR}]$ on the left-hand side. Then, a new form of the harmonic equation is obtained:

$$\left\{\left[K_{RR}\right] - \Omega^2\left[M_{RR}\right]\right\}\left\{\mathbf{u}_R\right\} = -\left\{\left[K_{RA}\right] - \Omega^2\left[M_{RA}\right]\right\}\left\{\mathbf{u}_A\right\}, \tag{4}$$

where $\mathbf{u}_R$ represents the unknown displacements and $\mathbf{u}_A$ the specified displacements. Here it is assumed that there are no forces applied on the R degrees of freedom. If such forces were applied, they would be included by adding a forcing term $\{\mathbf{f}_R\}$ to the right-hand side.

If the block is constructed from a polarized sample of piezoelectric material, e.g., Navy Type I,[8] rather than an elastic material, then one must specify the full matrix of elastic constants, rather than just the Young's modulus and Poisson's ratio. One must also specify the matrix of piezoelectric constants and dielectric constants, as well as the location of the electroded surfaces. If losses are included, these matrices are complex. Sample material matrices for the compliance constants $[s^E]$, piezoelectric strain constants, $[d]$, and dielectric constants $[\varepsilon^S]$ are given in appendix B.

For this example, suppose that the top surface of the block is an equipotential surface and that the bottom surface is a ground (see figure 3). Then, in addition to the 36 mechanical degrees of freedom, there are 4 electrical potential degrees of freedom corresponding to the electrical potentials at the midplane of the block (internal potentials), and 1 electrical potential degree of

freedom corresponding to the electroded top surface (external potential). The bottom surface has no free potentials since it is a ground.

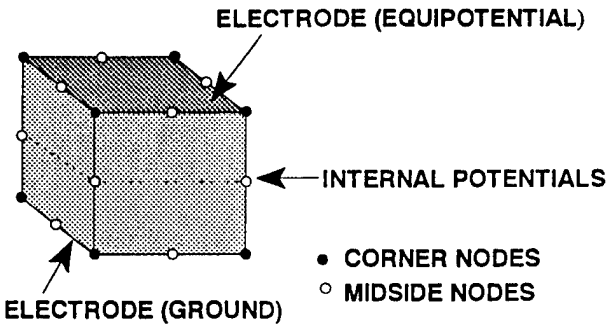


Figure 3. Piezoelectric Block with Electrodes

For piezoelectric structures, there is an elastic stiffness matrix $[K_{UU}]$ dependent on elastic moduli, a piezoelectric stiffness matrix $[K_{U\Phi}]$ dependent on piezoelectric constants, and a dielectric matrix $[K_{\Phi\Phi}]$ dependent on dielectric constants. For this example, $[K_{UU}]$ has dimensions (36 x 36), $[K_{U\Phi}]$ is [36 x (4+1)]=(36 x 5), and $[K_{\Phi\Phi}]$ is (5 x 5). One can write these matrices as partitions of a single matrix of dimensions (41 x 41) as follows:

$$[K] = \begin{bmatrix} K_{UU} & K_{U\Phi} \\ K_{U\Phi}^{t} & K_{\Phi\Phi} \end{bmatrix}, \tag{5}$$

where the subscript U indicates a mechanical or displacement degree of freedom and the subscript Φ corresponds to an electrical potential. Although this matrix is commonly referred to as the electromechanical stiffness matrix, this is in part a misnomer. The partitions of the matrix that relate to displacements $[K_{UU}]$ and $[K_{U\Phi}]$ truly describe the elastic and electromechanical stiffness. However, the purely electrical part of the matrix $[K_{\Phi\Phi}]$, which is a function of the dielectric constants of the material, is actually a compliance. The reason for this mixing of stiffness and compliance terms relates to the choice of constitutive equations upon which the finite element discretization is based.[9] For simplicity, the system matrix will be referred to as the stiffness matrix throughout this document.

The stiffness matrix, together with the mass matrix, can be used to compute all of the nodal displacements and potentials, both internal and external. However, one is not usually

interested in the values of the internal potentials (since they cannot be measured). Therefore, it is useful to separate the stiffness terms pertaining to internal potentials from those associated with external potentials, as in the following matrix:

$$[K] = \begin{bmatrix} K_{UU} & \mathbf{K}_{UI} & \mathbf{K}_{UE} \\ \mathbf{K}_{UI}^{t} & K_{II} & \mathbf{K}_{IE} \\ \mathbf{K}_{UE}^{t} & \mathbf{K}_{IE}^{t} & K_{EE} \end{bmatrix}, \tag{6}$$

where I corresponds to an internal potential and E to an external potential. The static equation ($\Omega = 0$) obtained with this partitioning of the stiffness matrix is as follows:

$$\begin{bmatrix} K_{UU} & \mathbf{K}_{UI} & \mathbf{K}_{UE} \\ \mathbf{K}_{UI}^{t} & K_{II} & \mathbf{K}_{IE} \\ \mathbf{K}_{UE}^{t} & \mathbf{K}_{IE}^{t} & K_{EE} \end{bmatrix} \begin{Bmatrix} \mathbf{u} \\ \boldsymbol{\phi}_{I} \\ \phi_{E} \end{Bmatrix} = \begin{Bmatrix} \mathbf{f} \\ -\mathbf{q}_{I} \\ -q_{E} \end{Bmatrix}, \tag{7}$$

where $\{\mathbf{q}_I\}$ and $\{\boldsymbol{\phi}_I\}$ are the nodal free charges and potentials, respectively, associated with the internal potentials, and q_E and ϕ_E are scalars associated with the electrodes (see appendix A).

The internal potentials are associated with zero nodal electrical charges ($\{\mathbf{q}_I\} = 0$), so one can "statically condense" them out of the solution, with no loss of accuracy. This is done by setting $\mathbf{q}_I$ to zero in equation (7) and then writing the internal potentials in terms of the displacements and external potential. First, the equation is rewritten for the static system as follows:

$$\begin{bmatrix} K_{UU} & \mathbf{K}_{UI} & \mathbf{K}_{UE} \\ \mathbf{K}_{UI}^{t} & K_{II} & \mathbf{K}_{IE} \\ \mathbf{K}_{UE}^{t} & \mathbf{K}_{IE}^{t} & K_{EE} \end{bmatrix} \begin{Bmatrix} \mathbf{u} \\ \boldsymbol{\phi}_{I} \\ \phi_{E} \end{Bmatrix} = \begin{Bmatrix} \mathbf{f} \\ \mathbf{0} \\ -q_{E} \end{Bmatrix}. \tag{8}$$

Equation (8) represents a set of three matrix equations, the second of which can be expanded as follows:

$$\{\mathbf{K}_{\mathbf{UI}}\}^t \{\mathbf{u}\} + [K_{II}]\{\boldsymbol{\phi}_{\mathbf{I}}\} + \{\mathbf{K}_{\mathbf{IE}}\}\phi_E = \{\mathbf{0}\} \ . \tag{9}$$

One can use equation (9) to write $\{\boldsymbol{\phi}_{\mathbf{I}}\}$ as a function of $\{\mathbf{u}\}$ and ϕ_E as follows:

$$\{\boldsymbol{\phi}_{\mathbf{I}}\} = -\{K_{II}\}^{-1}\{\{\mathbf{K}_{\mathbf{UI}}\}^t\{\mathbf{u}\} + \{\mathbf{K}_{\mathbf{IE}}\}\phi_E\} \ . \tag{10}$$

Then substitute this expression for $\{\boldsymbol{\phi}_{\mathbf{I}}\}$ into the first and third matrix equations of (8), group the like terms, and obtain modified expressions for the elastic, piezoelectric, and dielectric stiffness matrices with the internal potentials statically condensed as follows:

$$[K'_{UU}] = [K_{UU}] - \{\mathbf{K}_{\mathbf{UI}}\}[K_{II}]^{-1}\{\mathbf{K}_{\mathbf{UI}}\}^t, \tag{11}$$

$$\{\mathbf{K}'_{\mathbf{UE}}\} = \{\mathbf{K}_{\mathbf{UE}}\} - \{\mathbf{K}_{\mathbf{UI}}\}[K_{II}]^{-1}\{\mathbf{K}_{\mathbf{IE}}\} \ , \tag{12}$$

and

$$K'_{EE} = K_{EE} - \{\mathbf{K}_{\mathbf{IE}}\}^t[K_{II}]^{-1}\{\mathbf{K}_{\mathbf{IE}}\}. \tag{13}$$

$[K'_{UU}]$ still has dimensions (36 x 36), while $\{\mathbf{K}'_{\mathbf{UE}}\}$ is a (36 x 1) matrix, and K'_{EE} is a (1 x 1) matrix, or scalar, for our example. (If the block had more than one ungrounded electrode, this dimension would be greater than one, but the typical transducer has only one free external potential). Note that $[K'_{UU}]$ is hereafter referred to as the short-circuit stiffness matrix.

Using the modified matrices, one can solve for the behavior of a piezoelectric structure when it is used as either a driver (potential is an input) or a sensor (potential is an output). The general equation is as follows:

$$\begin{Bmatrix} [K'_{UU}] - \Omega^2\,[M] & \{\mathbf{K}'_{\mathbf{UE}}\} \\ \{\mathbf{K}'_{\mathbf{UE}}\}^t & K'_{EE} \end{Bmatrix} \begin{Bmatrix} \mathbf{u} \\ \phi_E \end{Bmatrix} = \begin{Bmatrix} \mathbf{f} \\ -q_E \end{Bmatrix}. \tag{14}$$

In the sensor mode, a harmonic force is applied and the displacements and the output voltage across the electrodes are computed. For this problem, the external charge is zero, and the harmonic equation is expressed as

$$\begin{Bmatrix} [K'_{UU}] - \Omega^2 [M] & \{\mathbf{K}'_{UE}\} \\ \{\mathbf{K}'_{UE}\}^t & K'_{EE} \end{Bmatrix} \begin{Bmatrix} \mathbf{u} \\ \phi_E \end{Bmatrix} = \begin{Bmatrix} \mathbf{f} \\ 0 \end{Bmatrix} . \tag{15}$$

Generally, one solves equation (15) in two parts. First, the second matrix equation is used to write ϕ_E as a function of $\mathbf{u}$, as follows:

$$\phi_E = - \{\mathbf{K}'_{UE}\}^t \{\mathbf{u}\}/K'_{EE} . \tag{16}$$

This expression is substituted for ϕ_E into the first matrix equation in equation (15), like terms are grouped, and a modified stiffness matrix is obtained as follows:

$$[K''_{UU}] = [K'_{UU}] - \{\mathbf{K}'_{UE}\} K'_{EE} \{\mathbf{K}'_{UE}\}^t . \tag{17}$$

$[K''_{UU}]$ is called the open-circuit stiffness matrix. Now one can solve for the displacements using equation (1), but substituting $[K''_{UU}]$ for $[K]$ as follows:

$$\{[K''_{UU}] - \Omega^2 [M]\} \{\mathbf{u}\} = \{\mathbf{f}\} . \tag{18}$$

Then the electrode potential is solved for using equation (16).

If the piezoelectric material is used as a driver, the potential is specified across the electrodes and the resulting displacements computed. One can use the first equation in (14), set the force to zero, move the potential term to the right-hand side, and solve for the displacements, as follows:

$$\{[K'_{UU}] - \Omega^2 [M]\}\{\mathbf{u}\} = -\{\mathbf{K}'_{UE}\} \phi_E . \tag{19}$$

Note that the short-circuit stiffness matrix has been used. Then, the computed displacements are substituted into the second part of equation (14) and the external charge computed, as follows:

$$q_E = -\{\mathbf{K}_{UE}'\}^t \{\mathbf{u}\} - K_{EE}' \phi_E. \quad (20)$$

Finally one can use the fact that the charge across the electrodes is related to the current by $q_E = I / (j\Omega)$, and the admittance is related to the current by $Y = I / \phi_E$, so that the admittance is related to the charge by $Y = (j\Omega q_E)/\phi_E$. The impedance can be obtained by inverting the complex admittance.

IN VACUO MODAL ANALYSIS

To determine the eigenvalues of the elastic system, return to equation (1), set the applied force to zero, and rewrite the equation as follows:

$$\{[K] - \omega_i^2 [M]\} \{\psi_i\} = \{\mathbf{0}\}, \quad (21)$$

where ω_i^2 is the set of eigenvalues (i = 1 to n, where n is the number of degrees of freedom in the system), and $\{\psi_i\}$ is the set of orthogonal eigenvectors, or modes. The natural frequencies of the system are found by taking the square root of the eigenvalues and dividing by (2π).

To determine the eigenvalues and eigenvectors of the piezoelectric system, one must specify one of two possible sets of electrical boundary conditions. Under the short-circuit condition, the potentials on both electrodes are set to zero, while in the open-circuit case, the charge is zero and the top electrode potential is free. The eigenvectors are virtually the same under the two conditions, but the eigenvalues are different. The short-circuit eigenvalues, commonly called resonance frequencies, represent the frequencies at which the mechanical system resonates; i.e., the frequencies for which the displacement response is a maximum. The open-circuit eigenvalues, or antiresonance frequencies, are the frequencies at which the input electrical impedance is a maximum.

One can apply the two electrical boundary conditions to find the eigenvalues and eigenvectors of the piezoelectric block under short- and open-circuit conditions. For a short-circuit analysis, set ϕ_E to zero and substitute the short-circuit stiffness matrix $[K_{UU}']$ for $[K]$ in equation (21), as follows:

$$\{[K_{UU}'] - \omega_i^2 [M]\} \{\psi_i\} = \{\mathbf{0}\}, \quad (22)$$

where $\{\psi_i\}$ is the set of short-circuit eigenvectors.

For the open-circuit case, ϕ_E is free (not zero), so $[K''_{UU}]$ rather than $[K'_{UU}]$ is used to represent the stiffness as follows:

$$\{[K''_{UU}] - \omega_i^2 [M]\} \{\psi_i\} = \{0\} , \tag{23}$$

where $\{\psi_i\}$ is the set of open-circuit eigenvectors.

Having obtained the short- and open-circuit frequencies, one can compute the *in vacuo* electromechanical coupling constant k for each mode, using the following relation:

$$k^2 = \frac{\omega_a^2 - \omega_r^2}{\omega_a^2}, \tag{24}$$

where ω_a and ω_r are the radian open-circuit and short-circuit frequencies, respectively.

At this point, one is able to solve for the eigenvalues and eigenvectors of an elastic or piezoelectric structure, and for the harmonic response with a specified force, displacement, or electrode potential, assuming there is no fluid present. The description of an enclosed or surrounding fluid domain is given in section 3.

3. BOUNDARY ELEMENT MODEL

BACKGROUND

Using the boundary element method, the wetted surface of the structure is discretized into a continuous set of two-dimensional elements and a corresponding set of nodes. These elements may lie on the inside, if the fluid is in the interior of the structure, or on the outside, if the fluid surrounds the structure, or both.

To continue with the block example, suppose that all six sides of the block are exposed to a surrounding fluid. Assume that the length of a side is much smaller than the acoustic wavelength $\lambda = c/f$, where c is the speed of sound in the fluid and f is the frequency of operation. Then, one can construct the boundary element model using six quadrilateral elements, one on each face (see figure 4). Each quadrilateral boundary element is defined by eight nodes. Because we are assuming a patch BE code, the surface pressures and velocities are prescribed at the area centroid of each patch or element. The boundary element model can be used alone to solve two kinds of problems. The first is a case in which one knows the velocity distribution of the

fluid-loaded structure and wants to determine the pressure in the fluid. The second is a case in which a plane wave is incident on the block , which is rigid, and one wants to determine the surface pressures or the scattered field pressures.

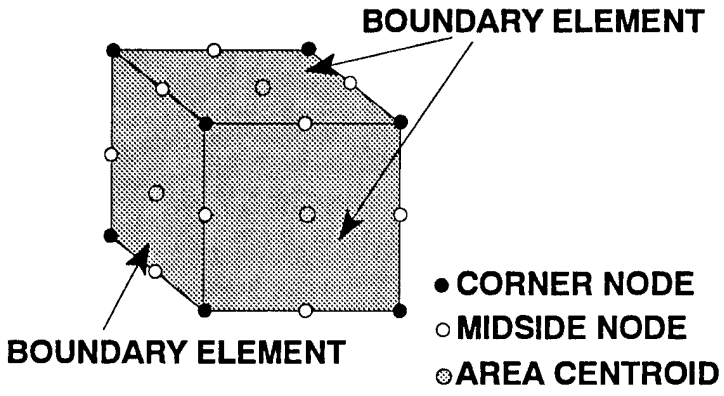


Figure 4. Boundary Elements on Wetted Surface

RADIATION PROBLEM

Suppose one knows the velocities at the centroid of each side of the block, and that one wants to compute the pressures in the fluid surrounding the block. It is not necessary to know what is on the inside of the block, because the surface velocities are known. One has to specify the density ρ and the speed of sound c of the fluid. Then the boundary element program computes two matrices, designated in CHIEF as $[A]$ and $[B]$, that depend on the surface geometry, the fluid properties, and the frequency. These matrices are used to compute the surface pressures $\{\mathbf{p}\}$ for a given normal velocity distribution $\{\mathbf{v}_N\}$ as follows:

$$[A]\{\mathbf{p}\} = [B]\{\mathbf{v}_N\}. \tag{25}$$

Equation (25) has a unique solution, if $[A]$ is well conditioned. However, there are certain frequencies at which the solution is nonunique. This problem can be avoided using any of several

techniques. In the CHIEF code, the user specifies the coordinates of some number of interior points, that are used to add additional constraints on the pressure. Then the system becomes overdetermined and is solved using the Householder method.[10]

The frequencies at which the solution becomes nonunique correspond to the eigenvalues of the interior cavity with pressure-release boundaries (pressure equals zero on the surface). Some boundary element codes automatically take care of this problem; while in the CHIEF code, it is up to the user to specify interior points or not, depending on whether the operating frequency is thought to lie near an interior eigenfrequency. If there is a spherical surface, one can compute these frequencies by setting the acoustic wavelength equal to the diameter of the sphere, $\lambda = d$, to get the following equation:

$$f_{res} = \frac{nc}{d} \; ; n=1,2,3... \; , \tag{26}$$

where d is the diameter of the sphere and c is the speed of sound in the fluid. Unfortunately, the problem frequencies for a realistic geometry cannot be computed, so another way must be found to know whether one needs to specify interior points. One could take the approach of always providing interior points, but there is a way to determine when it is really necessary. This test will be presented in section 4.

SCATTERING PROBLEM

The second type of problem that can be solved using the boundary element model alone is the scattering of a plane acoustic wave by a rigid surface. If one specifies the magnitude and direction of the incident wavefront, the boundary element code can compute the incident surface pressure $\{\mathbf{p}_{inc}\}$ on each element. Then the code can compute the resulting surface pressures using the following equation:

$$[A]\{\mathbf{p}\} = \{\mathbf{p}_{inc}\}. \tag{27}$$

One can then solve for rigid scattered pressures in the nearfields and farfields.

4. COUPLED FINITE ELEMENT/BOUNDARY ELEMENT MODEL

BACKGROUND

The finite element and boundary element models can be used individually to solve certain problems. However, most sonar problems require the two models to be combined because of the interest in the in-fluid behavior, and one usually does not have enough information about the in-fluid surface velocities of the structure to predict the field pressures with the boundary element model alone. To couple the two models, the elements of the boundary element model must have a one-to-one correspondence with the fluid-loaded surfaces of the finite element model. For our example, each face of the single finite element is represented by a boundary element, so there are six boundary elements, the nodes of which have the same coordinates as the corresponding nodes of the finite element.

For the coupling problem, the fluid is characterized by a complex matrix called the mutual coupling matrix $[Z]$. This matrix contains the same information as the $[A]$ and $[B]$ matrices, but in a form that directly describes the interaction through the fluid of the various elements on the wetted surface, i.e., it is a measure of the self and mutual impedances of the surface elements. The mutual coupling matrix is a function of the wetted surface geometry, the fluid properties, and the frequency of operation. The matrix does not depend on the material properties of the structure or on the velocity distribution on the surface. Each matrix element z_{ij} quantifies the force on the area centroid of surface element i, caused by a unit-normal velocity on surface element j, when all other surface elements are rigid, as given by the following equation:

$$z_{ij} = [A^{-1}B]_{ij} S_i = \left. \frac{p_i S_i}{v_{Nj}} \right|_{\substack{(v_{Nj}=1) \\ (v_{Nk\neq j}=0)}}, \tag{28}$$

where p_i is the pressure, S_i is the area of surface i, and $\overline{v}_{Nj}$ is the normal velocity of surface j. For this example, the dimensions of $[Z]$ are (6 x 6), with one row and one column corresponding to each boundary element.

As was the case for the $[A]$ and $[B]$ matrices discussed earlier, the computation of the mutual coupling matrix $[Z]$ is subject to problems at particular frequencies that correspond to the interior eigenfrequencies. What is needed is a way to determine whether or not one happens to be operating at one of these frequencies. One can determine this by using the fact that the mutual coupling matrix is symmetric, due to reciprocity.[11] Symmetry implies that $z_{ij} = z_{ji}$, i.e., one should get the same value when one puts a unit velocity on surface j and computes the pressure on

surface i, as when one puts a unit velocity on surface i and computes the pressure on surface j. At a frequency for which the solution is nonunique, the $[Z]$ matrix will have a high degree of asymmetry. Therefore, assuming that the boundary element model is error free, compute the asymmetry of the matrix to be sure there is not a need to include interior points. This concept is demonstrated in the example problem to follow.

The real part $[R]$ of the mutual coupling matrix describes the effects of radiation damping, which influences the structural vibration in the same way as hysteresis or material damping. Radiation damping primarily lowers the vibration level in the vicinity of resonance. For interior fluids, there is no loss mechanism, so the real part of the $[Z]$ matrix is zero. For an exterior fluid, the radiation damping represents the portion of energy that radiates to the acoustic farfield. The imaginary part $[X]$, when divided by ω, represents the mass loading of the fluid. It is nonzero for both interior and exterior fluids. The primary effect of the radiation mass is to lower the resonance frequency of the structure relative to its frequency in the absence of fluid.

The coupled problem can be solved in two ways. One method (used at the Naval Undersea Warfare Center, Underwater Sound Reference Detachment, Orlando, FL) is to import the $[Z]$ matrix into the FE code (ATILA[12]), and solve for the in-fluid displacements and velocities directly. In this case, the FE model must include not only the elements representing the structure, but also elements describing the wetted surface. The latter elements must have coordinates identical to those in the BE model, and they must appear in the same order. Then each row and column of the $[Z]$ matrix pertains to the same element in both models (FE and BE). The second method is less direct, but does not require the FE code to have any special capabilities other than writing out the stiffness, mass, and compatibility matrices. Using this approach, $[K]$ and $[M]$ are computed and written out by the FE code. The compatibility matrix is also computed by the FE code, by applying a unit pressure to successive surfaces and retaining the resulting nodal forces in a file. (This procedure is described in the following section.) Then $[Z]$ is computed by the BE code and stored in a file. Finally, all of these files are read by a FORTRAN program that combines the matrices in the appropriate manner for a particular analysis, and then solves the coupled fluid/structure equation. The various equations for these analyses are provided later in this document.

COMPATIBILITY MATRIX

The compatibility matrix is essentially a translator between pressures or normal velocities at the centroid of a wetted surface and forces or displacements at the nodes on the same surface. One can define each column of the compatibility matrix as the set of consistent nodal forces that corresponds to a unit pressure applied on a particular element on the wetted surface, normalized by the area of the surface. Alternatively, one can say that each element of the $[C]$ matrix C_{ij}, is the fraction of the total force on boundary element j applied in the direction of degree of freedom i, or $C_{ij} = F_{dof\ i}/F_{total\ bej} = F_{dof\ i}/(p_{be\ j}S_{be\ j})$. Using these definitions, one can generate the compatibility with any finite element code.

For example, consider the wetted surface shown in figure 5. The surface is divided into four boundary elements, each of which has four nodes (corner nodes only) and area, $S = 4.0\ m^2$. Suppose that the nodes are free to move only in the direction normal to the surface, so that it is not necessary to deal with the vector components of the velocity. Then, each node has 1 DOF, and the 9 nodes have a total of 9 DOFs, numbered as shown in figure 5. The compatibility matrix will have nine rows, one for each degree of freedom, and four columns, one for each boundary element. Now, apply a negative unit pressure in pascals to boundary element 1, and zero pressure to the other elements, and look at the consistent nodal forces computed by the finite element code. The total force corresponding to the unit pressure for this example is 4.0 N. A four-noded finite element will have the forces evenly distributed among the nodes as shown in figure 5. Therefore, the pressure on surface 1 will give a normal force of 1.0 N at each of the nodes on element 1. Because each node has one degree of freedom, and these are all in the direction normal to the surface, the force applied in the direction of the degree of freedom is equal to the total force at the node or 1.0 N. Each element of $[C]$ is the force in the direction of a particular degree of freedom divided by the total force on the element, or $C_{ij} = 1.0\ N/4.0\ N = 0.25$. Then, the first column of $[C]$ has nonzero elements in rows 1, 2, 4, and 5, as follows:

$$[0.25\ \ 0.25\ \ 0.0\ \ 0.25\ \ 0.25\ \ 0.0\ \ 0.0\ \ 0.0\ \ 0.0]^t.$$

(Note that this is the transpose of the column.) If one then applies a negative unit pressure to surface 2, one obtains the second column of $[C]$:

$$[0.0\ \ 0.25\ \ 0.25\ \ 0.0\ \ 0.25\ \ 0.25\ \ 0.0\ \ 0.0\ \ 0.0]^t,$$

with forces applied in DOF 2, 3, 5, and 6. Proceed with the last two columns to obtain the complete $[C]$ matrix:

$$\begin{bmatrix} 0.25 & 0.0 & 0.0 & 0.0 \\ 0.25 & 0.25 & 0.0 & 0.0 \\ 0.0 & 0.25 & 0.0 & 0.0 \\ 0.25 & 0.0 & 0.25 & 0.0 \\ 0.25 & 0.25 & 0.25 & 0.25 \\ 0.0 & 0.25 & 0.0 & 0.25 \\ 0.0 & 0.0 & 0.25 & 0.0 \\ 0.0 & 0.0 & 0.25 & 0.25 \\ 0.0 & 0.0 & 0.0 & 0.25 \end{bmatrix}.$$

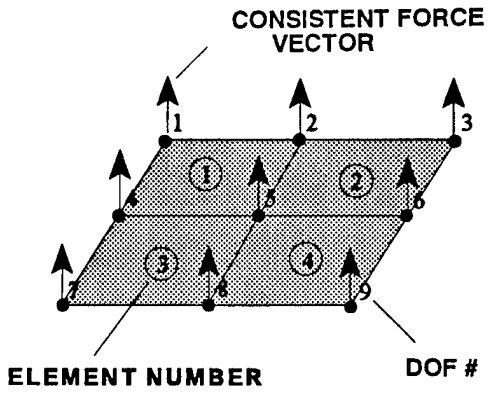


Figure 5. Consistent Forces

Note that the sum of each column is one, and that the only row that has a nonzero value in each column corresponds to DOF 5, which in turn corresponds to node 5, the only node that touches every boundary element. In this example, 4-noded quadrilateral elements were used, resulting in equal division of the force among the degrees of freedom of the nodes on a given element. Had one used 8-noded elements, this would not have been the case, and the compatibility matrix would have involved fractions other than 1/4. This will be demonstrated in the example problem later this document.

IN-FLUID HARMONIC ANALYSIS

Radiation Problem

Because the mutual coupling matrix describes the relationships between surface pressures and surface velocities, one can use it to compute the surface pressures at a specified frequency for a known surface velocity distribution. For example, if one is interested in the surface pressures on a pulsating sphere, one knows that the normal velocities $\{\mathbf{v}_N\}$ are the same everywhere on the surface. One can compute the mutual coupling matrix at the frequency of interest and multiply it by the surface velocities to obtain the surface pressures, as follows:

$$\{\mathbf{p}\} = [Z]\{\mathbf{v}_N\} \ . \tag{29}$$

Because generally the velocities are unknown, one needs a way to compute the velocities of the structure in the presence of the fluid. Then the computed velocities can be used to obtain surface pressures and field pressures. To compute the surface velocities, one must combine the boundary element matrix with the finite element matrices and solve the coupled system of equations. In combining these matrices, one must account for the fact that the boundary element quantities are prescribed at the centroid of a patch, and the finite element quantities are prescribed at nodes. To do this, one can either translate the nodal quantities into equivalent patch values, or translate the patch quantities into nodal values. Here the latter approach has been chosen; the former approach is described in reference 5.

Using the nodal approach, the general equation describing the coupling between elastic finite elements and fluid boundary elements in a harmonic analysis is as follows:

$$\{[K] - \Omega^2 [M] + j\Omega [C][Z(\Omega)][C]^t\} \{\mathbf{u}\} = \{\mathbf{f}\} , \tag{30}$$

where $[C]$ is the compatibility matrix, Ω is the excitation frequency in radians, and $\{\mathbf{f}\}$ is the vector of applied mechanical forces. Physically, the term containing $[Z]$, when multiplied by $\{\mathbf{u}\}$, represents the consistent nodal forces due to the fluid loading. The compatibility matrix interprets between nodal quantities, i.e., forces or displacements, on the elastic structure and centroidal

quantities, i.e., forces or normal velocities, on the wetted surface. The dimensions of $[C]$ are {ndof x nsurf} where nsurf is the number of boundary elements in the model. For this example (see figures 3 and 4), $[C]$ has dimensions {36 x 6}. Each component of the matrix C_{ij} describes the nodal force on node i due to a unit pressure applied on surface j.

Equation (30) results in the in-fluid surface displacements of the structure as a function of frequency. One can compute the normal surface velocities using the following relation:

$$\{\mathbf{v}_N\} = j\omega[C]^t\{\mathbf{u}\}. \tag{31}$$

If one is interested in the field pressures corresponding to this velocity distribution, supply the velocity values to the boundary element code, which will then compute the pressure in the nearfield or farfield.

For a piezoelectric structure, the equation for the harmonic response due to an applied potential on the electrode is:

$$\{[K'_{UU}] - \Omega^2\,[M] + j\Omega[C][Z][C]^t\}\,\{\mathbf{u}\} = -\{\mathbf{K}'_{UE}\}\,\phi_E\,. \tag{32}$$

The displacements can again be used to compute the normal velocities of the surface through equation (31), and the electrical charge through (20). The electrical admittance can then be calculated from the charge as described in the text that immediately follows (20).

The procedure for computing the in-fluid displacements and velocities and the resulting surface and field pressures, is performed in four steps. In the first step, the boundary element code is used to compute the complex mutual coupling matrix, $[Z]$, for each frequency of interest. For example, if we are interested in frequencies between 1000 and 1100 Hz, in 10 Hz intervals, one must compute $[Z]$ at 1000 Hz, 1010 Hz, and so on. This can be a lengthy process, especially for wide-band problems, but there is a way to make the process more efficient without giving up too much in the way of accuracy. The $[Z]$ matrix varies slowly with frequency, so one can do a true computation of the matrix at widely spaced frequencies and generate the matrix at intermediate frequencies using interpolation.[13] This capability is available in the XCID program.[6]

The second step in the radiation problem is to use the finite element code to generate the stiffness, mass, and compatibility matrices. These matrices can be combined with the mutual coupling matrix in an external program and use equations (30) or (32) to compute the in-fluid displacements, and equation (31) to compute the normal velocities at each frequency of interest. In the fourth step, the boundary element code with the velocities at each frequency have been provided, and the surface pressures, nearfield pressures, and farfield pressures have been computed. One can also compute the source level and directivity index. A flowchart for the coupled radiation procedure is provided later, in section 5, figure 6.

Scattering Problem

If the excitation is an incident plane acoustic wave, the analysis becomes a bit more complicated. The field pressure due to an incident wave has two components. One component is contributed by the rigid scattering from the surface, and the other is contributed by the elastic scattering. The elastic scattering can also be viewed as direct radiation from the vibrating surface, the motion of which is caused by the incident pressure of the acoustic wave. Following the analysis of Junger and Feit,[14] the problem can be solved in three steps. First, CHIEF computes the surface pressures on the rigid structure $\{\mathbf{p}_{sr}\}$. If one is interested in field pressures, one must also request the rigid scattered pressures $\{\mathbf{p}_r\}$. (It is also necessary to compute the $[Z]$ matrix in this step, but this is not unique to the scattering analysis.) Then determine the equivalent nodal forces on the surface from the computed surface pressures, using the relation:

$$\{\mathbf{f}\} = -[C]\{\mathbf{p}_{sr}\} \; . \tag{33}$$

Next apply these forces to the surface nodes in the finite element model, and compute the resulting in-fluid displacements, using the following equation:

$$\left\{ [K] - \Omega^2 [M] + j\Omega[C][Z][C]^t \right\} \{\mathbf{u}\} = -[C]\{\mathbf{p}_{sr}\} \; . \tag{34}$$

Then the normal surface velocities are computed using equation (31). Finally, the vector of normal velocities is supplied to the boundary element code, and the elastic scattered pressures $\{\mathbf{p}_e\}$ computed. The total field pressures, excluding the incident pressure, are obtained by summing $\{\mathbf{p}_r\}$ and $\{\mathbf{p}_e\}$. The sum may then be used to compute the target strength of the elastic structure, if desired.

In the case of scattering from a piezoelectric structure, the active (piezoelectric) material is generally used as a sensor, rather than as a driver. Then not only the velocity of the surface is computed, but also the voltage output from the sensor. To do this, one uses a slightly different form of the equation as follows:

$$\left\{ \left([K''_{UU}] - \Omega^2 [M] \right) + j\Omega[C][Z][C]^t \right\} \{\mathbf{u}\} = -[C]\{\mathbf{p}_{sr}\} \; , \tag{35}$$

where $[K''_{UU}]$ is the open-circuit stiffness matrix, equation (17), and ϕ_E is the sensor voltage. After computing the displacements, equation (16) can be used to determine the sensor voltage.

Note that the only significant difference between this procedure and the one for a radiation problem is that in the initial boundary element run the rigid scattered surface pressures are computed in addition to the mutual coupling matrix. These pressures become the excitation, replacing the mechanical or electrical excitation in the radiation problem. A flowchart for the coupled scattering procedure is provided in section 5, figure 7.

IN-FLUID MODAL ANALYSIS

Recall that for the *in vacuo* structure (FE model alone), it was possible to write the equation for a modal solution, equation (21), by setting the right-hand side of the harmonic equation, equation (1), to zero. Then, the known quantities were the stiffness and mass matrices, and the unknown quantities were the eigenvalues and eigenvectors. The form of equation (21) is appropriate for an eigensolution because $[K]$ and $[M]$ are independent of frequency. If one examines equation (30) for the in-fluid harmonic solution, one sees that the stiffness and mass matrices appear the same as in equation (1), but there is an additional term representing the fluid loading:

$$\text{fluid load} = \{j\Omega\ [C][Z(\Omega)][C]^t\}\ . \tag{36}$$

This term is frequency dependent, so the left hand side of equation (30) cannot be made to fit the form of the standard eigenvalue equation.

Until recently, there did not exist a method for determining the in-fluid eigenvalues and eigenvectors for a FE/BE model. In-fluid eigenfrequencies were obtained only under the assumption that the fluid is incompressible.[15] Then, the sound speed becomes infinite and the wavenumber $k = \omega/c$ is zero. Therefore, the fluid loading was computed at a frequency of zero. This approximation can work well for some very low-frequency transducers, but it can lead to very large errors in many cases. More commonly, the in-fluid resonance frequencies were determined instead, by performing successive harmonic analyses until the peak in the harmonic response was found.

Recently, the authors proposed a method for finding the in-fluid eigenvalues without assuming incompressibility of the fluid.[16] The basic idea behind this method is to modify the structural matrices in equation (21) by adding successive approximate radiation damping values to the stiffness matrix and approximate radiation mass values to the mass matrix. It is approximate because the radiation load, quantified by the mutual coupling matrix, is frequency dependent so one must specify a frequency at which to compute the matrix. Because it is uncertain which frequency to specify, the best approach is to use the only known frequency, i.e., the *in vacuo*

eigenfrequency, which we will denote as ω_o. One can compute the mutual coupling matrix at the *in vacuo* frequency, modify the structural matrices, and compute the eigenvalues of the resulting equation:

$$\left\{([K] + j\omega_o[C][R(\omega_o)][C]^t) - \omega_i^2\left([M] + \frac{[C][X(\omega_o)][C]^t}{\omega_o} \right)\right\}\{\psi_i\} = \{0\}, \tag{37}$$

where ω_i^2 and $\{\psi_i\}$ are the in-fluid eigenvalues and eigenvectors, respectively, for the given fluid loading. Here the real and imaginary parts of the $[Z]$ matrix have been separated. For simplicity, equation (37) can be rewritten as

$$\{[K^f] - \omega_i^2[M^f]\}\{\psi_i\} = \{0\}, \tag{38}$$

where

$$[K^f] = [K] + j\omega_o[C][R(\omega_o)][C]^t, \tag{39}$$

and

$$[M^f] = [M] + \frac{[C][X(\omega_o)][C]^t}{\omega_o}. \tag{40}$$

In practice, it is generally not necessary to include the radiation damping because its effect on the in-fluid eigenfrequency is small compared with that of the radiation mass. The serendipitous result of neglecting the radiation damping is that equation (37) becomes purely real, greatly reducing the computation time.

Note that $j\omega_o[C][R(\omega_o)][C]^t$ and $[C][X(\omega_o)][C]^t/\omega_o$ each comprises a matrix of constants; therefore, equation (37) is a mathematically valid eigenvalue equation that can be solved for eigenvalues and eigenvectors. However, the fluid load was computed at the *in vacuo* eigenfrequency, so it does not represent the true fluid load, i.e., the load at the in-fluid eigenfrequency. For typical sonar problems, the fluid load computed at the *in vacuo* frequency is greater than that at the in-fluid eigenfrequency, so it is likely the radiation damping and mass have been overestimated. Furthermore, even if we knew the in-fluid resonance frequency of the mode of interest and computed the fluid load at that frequency, equation (37) would only give us the correct eigenvector for that particular mode. The others modes have different in-fluid resonance

frequencies, so they would still not have the correct fluid load. For this reason, this method can be used only for determining one in-fluid eigenvalue at a time. For practical design problems, this is not a serious limitation because the designer is generally only interested in one mode. And, of course, the procedure can be performed again for each additional mode of interest.

The problem remains of finding the correct value of the in-fluid eigenfrequency for the particular mode of interest. To do this, one begins with the *in vacuo* eigenfrequency, as stated above, compute the $[Z]$ matrix at this frequency, modify the structural mass matrix (and the stiffness matrix, if desired), and solve the resulting eigensystem. If this particular mode remains uncoupled under fluid loading, one will find that there is a mode in the set of computed in-fluid eigenvectors that closely resembles the *in vacuo* mode shape of interest. Set ω_o equal to the frequency corresponding to this mode, recompute the $[Z]$ matrix at the new value of ω_o, and so on, until two consecutive in-fluid eigenvalue computations match for the mode of interest, within a specified tolerance.

It should be noted that modal preservation, that is, the absence of modal coupling,[17, 18] is a requirement for this method to be meaningful. If two or more *in vacuo* modes contribute to the same peak in the in-fluid harmonic response, then there is not a unique eigenfrequency corresponding to that peak.

To find the in-fluid eigenvalues of a piezoelectric structure acting as a driver, substitute the short-circuit stiffness matrix from equation (11) for the elastic stiffness matrix in equation (37), so that $[K^f]$ in equation (39) becomes:

$$[K^f] = [K'_{UU}] + j\omega_o[C][R(\omega_o)][C]^t. \tag{41}$$

Again, if one neglects the effects of radiation damping, simply use the short-circuit stiffness matrix alone. By using the short-circuit, rather than the open-circuit stiffness matrix, one obtains the frequency at which the harmonic displacement response peaks when the device is excited by an applied electrical potential. One can also find the in-fluid open-circuit frequency, corresponding to the harmonic displacement peak when the device is acting as a sensor, by substituting $[K''_{UU}]$ for $[K'_{UU}]$, as follows:

$$[K^f] = [K''_{UU}] + j\omega_o[C][R(\omega_o)][C]^t. \tag{42}$$

A flowchart for the coupled modal procedure is provided in section 5, figure 8.

INTERIOR AND EXTERIOR FLUIDS

For all of the analyses presented thus far, it has been assumed that the fluid medium is either on the interior or the exterior of the structure, but not on both. However, many sonar transducers are oil-filled, and of course, they operate in seawater. To model this scenario, one can use any of the equations presented in the previous section for a coupled analysis, but one must make a modification to the fluid loading term.

To begin with, the boundary element code is designed to describe only one fluid domain. So, there must be two separate boundary element models; one describing the geometry of the interior surface and the properties of the interior fluid, and the other describing the same for the exterior. Using each of the models in turn, compute the mutual coupling matrices, $[Z_{int}]$ and $[Z_{ext}]$, for the interior and exterior domains, respectively. Note that if one is performing a scattering analysis, the pressures on the rigid surface are computed in the exterior domain only.

Then, combine the two matrices into a single mutual coupling matrix as follows:

$$[Z] = \begin{bmatrix} Z_{int} & 0 \\ 0 & Z_{ext} \end{bmatrix}. \tag{43}$$

The dimensions of the combined $[Z]$ matrix is (nsurf x nsurf), where nsurf is the total of the interior and exterior boundary elements. The reason that the off-diagonal partitions are zero is that the two domains have no acoustic interaction with each other. For a scattering analysis, combine the set of surface pressures on the exterior surface with a null set as follows:

$$\{\mathbf{p}_{sr}\} = \begin{Bmatrix} 0 \\ \mathbf{p}_{sr_{ext}} \end{Bmatrix}. \tag{44}$$

The dimensions of the combined pressure vector are (nsurf x 1). The null set describes the fact that the interior of the structure does not "see" the incident plane wave directly. While the final computation of the exterior fluid pressure is a combination of the rigid and elastic scattering, the interior fluid pressure comes from the elastic vibration alone.

In the case of two fluid domains, compute the compatibility matrix $[C]$ in exactly the same way as for a single fluid domain (appendix A), except that now the dimensions are (ndof x nsurf), where nsurf is the total of the interior and exterior wetted surfaces. Once the appropriate $[Z]$, $\{\mathbf{p}_{sr}\}$, and $[C]$ matrices are obtained, one can perform any coupled FE/BE analysis, including radiation and scattering. Compute the surface velocities using the coupled equations, and input

the interior surface velocities into the interior BE model, or the exterior velocities into the exterior model, or both.

5. SUMMARY OF EQUATIONS AND PROCEDURES

This section summarizes the equations and procedures for the various *in vacuo* and in-fluid analyses. Table 1 provides a cross reference of solutions with equation numbers from the text. Figures 6 and 7 are the flowcharts for the coupled radiation and scattering procedures, respectively. Figure 8 is the flowchart for the iterative in-fluid modal procedure.

Table 1. Cross Reference of Equation Numbers

Medium	Analysis	Excitation	Mat'ls	Model	Output	Eq. #
Vacuum	H	Mechanical force	E	FE	$\{\mathbf{u}\}$	(1)
"	H	Specified displacement	E	FE	$\{\mathbf{u}\}$	(4)
"	H(OC)	Mechanical force	E/P	FE	$\{\mathbf{u}\}, \phi_E$	(18),(16)
"	H(SC)	Electrical potential	E/P	FE	$\{\mathbf{u}\}, q_E$	(19),(20)
"	M	None	E	FE	$\omega_i^2, \boldsymbol{\psi}_i$	(21)
"	M (SC)	None	E/P	FE	$\omega_i^2, \boldsymbol{\psi}_i$	(22)
"	M (OC)	None	E/P	FE	$\omega_i^2, \boldsymbol{\psi}_i$	(23)
Fluid	H	Specified velocity	NA	BE	$\{\mathbf{p}\}$	(25)
"	H	Acoustic pressure, $\overline{v}_N=0$	NA	BE	$\{\mathbf{p}\}$	(27)
"	H	Mechanical force	E	FE/BE	$\{\mathbf{u}\},(\mathbf{v}_N\},\{\mathbf{p}\}$	(30),(31),(25)
"	H(SC)	Electrical potential	E/P	FE/BE	$\{\mathbf{u}\},(\mathbf{v}_N\},\{\mathbf{p}\}$	(32),(31),(25)
"	H	Acoustic pressure	E	FE/BE	$\{\mathbf{u}\},(\mathbf{v}_N\},\{\mathbf{p}\}$	(34),(31),(25)
"	H(OC)	Acoustic pressure	E/P	FE/BE	$\{\mathbf{u}\},(\mathbf{v}_N\},\{\mathbf{p}\}$	(35),(31),(25)
"	M*	None	E	FE/BE	$\omega_i^2, \boldsymbol{\psi}_i$	(38),(39),(40)
"	M (SC)*	None	E/P	FE/BE	$\omega_i^2, \boldsymbol{\psi}_i$	(38),(40),(41)
"	M(OC)*	None	E/P	FE/BE	$\omega_i^2, \boldsymbol{\psi}_i$	(38),(40),(42)

*Iterative procedure

H = harmonic, M = modal, SC = short circuit, OC = open circuit, E = elastic, P = piezoelectric, NA = not applicable

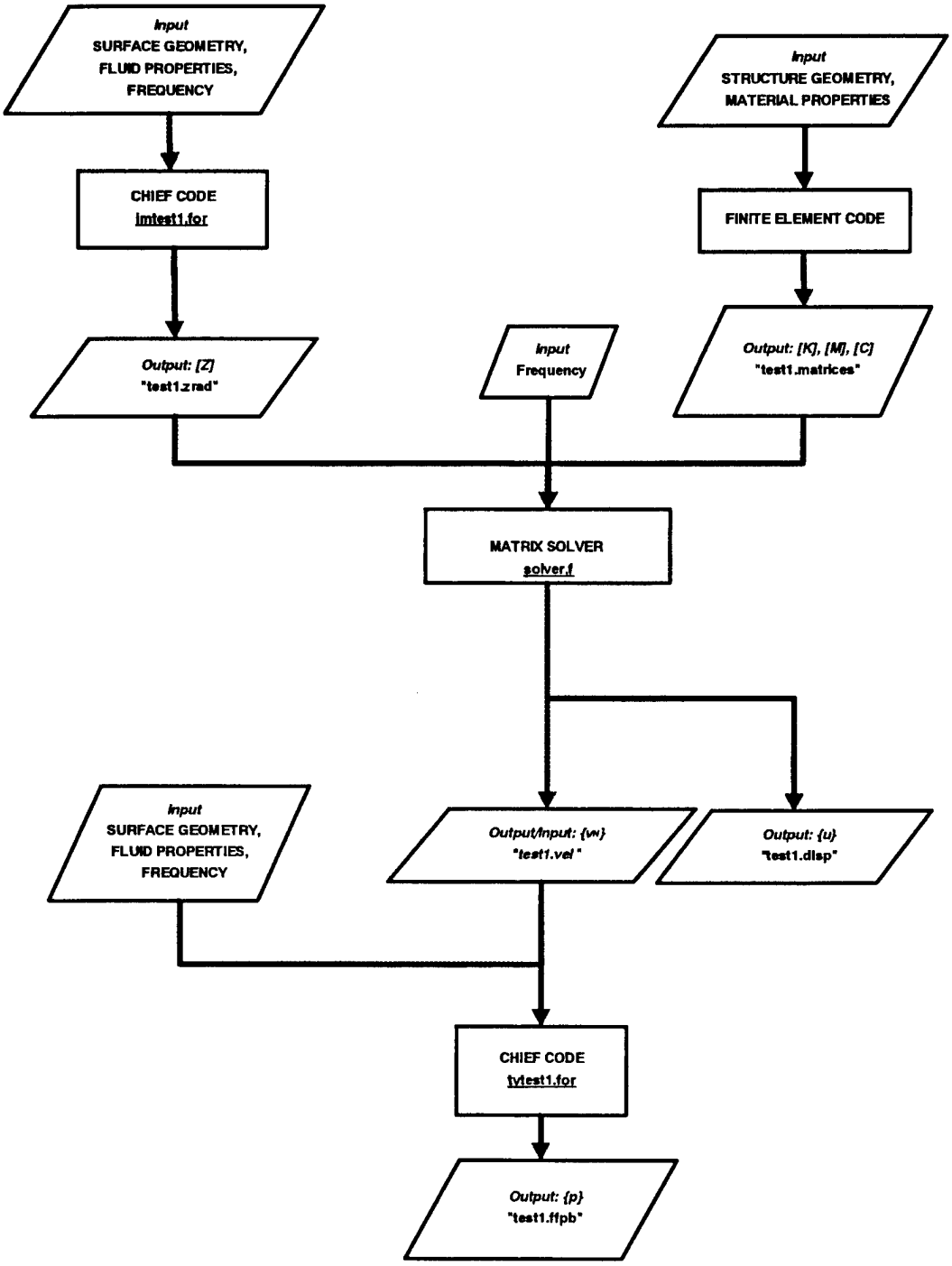


Figure 6. Flowchart for In-Fluid Radiation Procedure

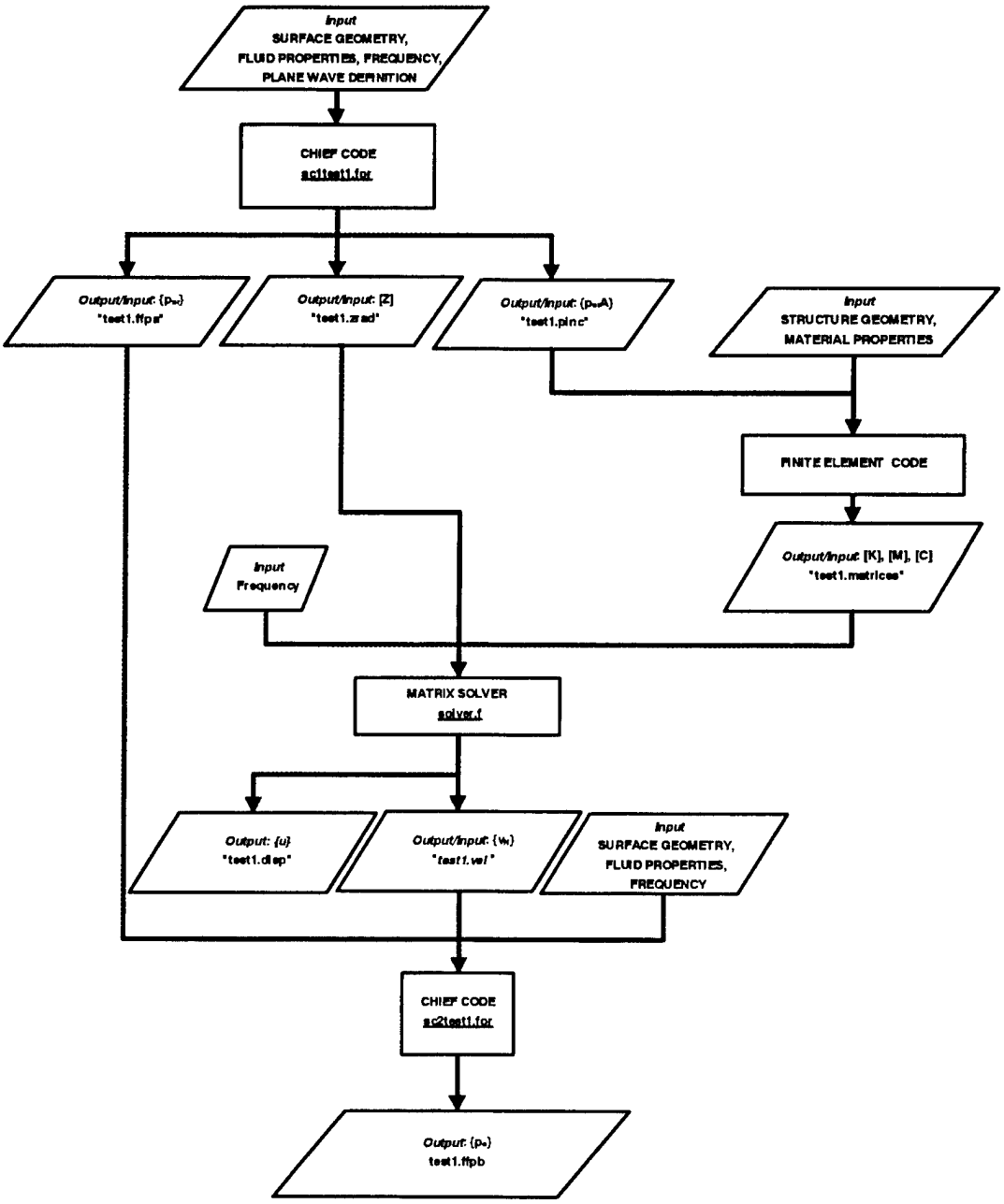


Figure 7. Flowchart for In-Fluid Scattering Procedure

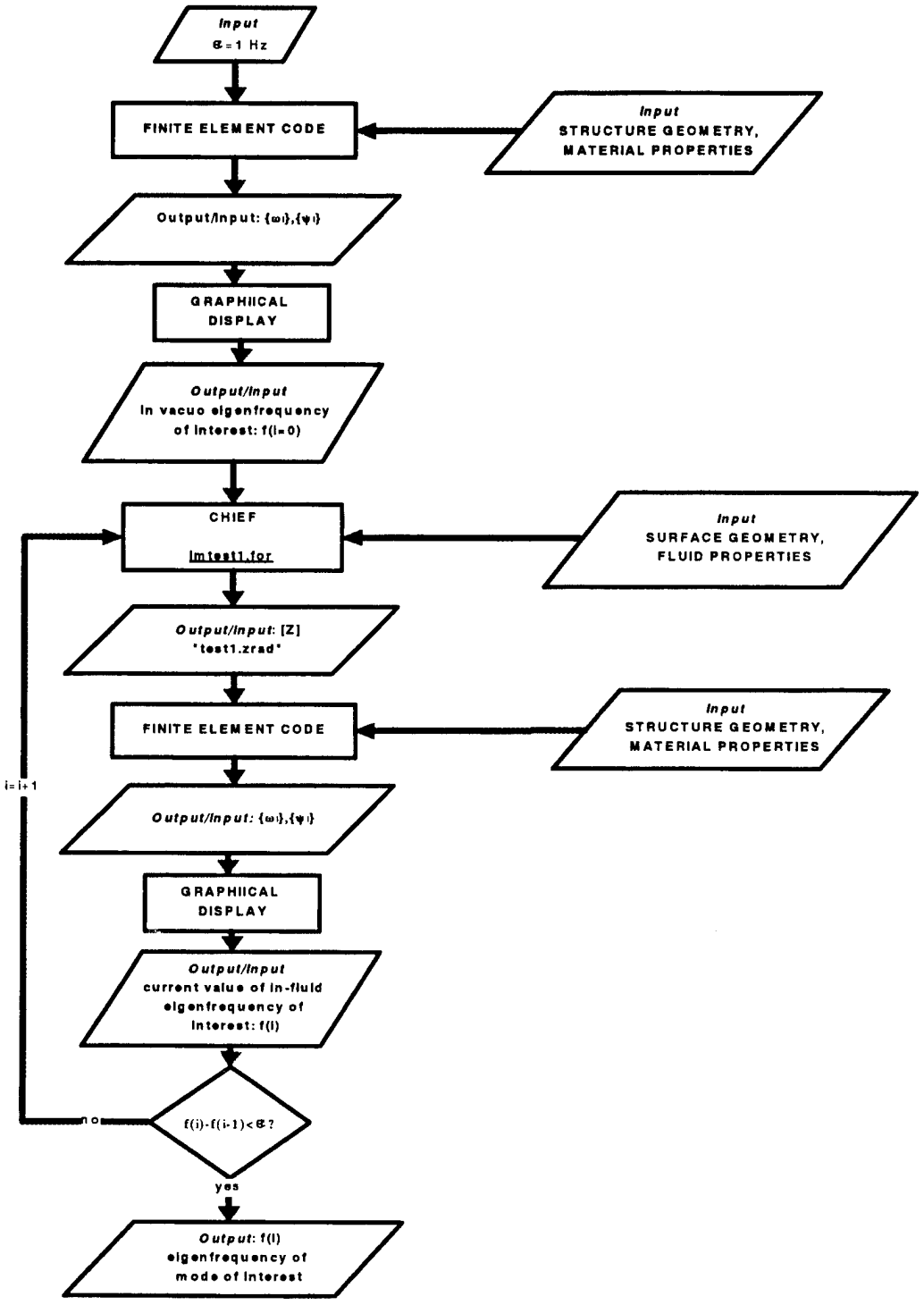


Figure 8. Flowchart for In-Fluid Modal Analysis

6. POTENTIAL ERRORS IN THE COUPLING PROCEDURE

BACKGROUND

Generating a finite element or boundary element model is generally not a difficult problem. The difficulty lies in creating a model that is both representative of the physical structure and free of errors. The first problem can be addressed only by careful comparison of the numerical model results with those obtained from measurements and existing analytic models. The second problem, that of generating an error-free coupled FE/BE model, can be addressed using a number of techniques that will be described in this section. These techniques have been developed by Navy modelers over the last several years in response to problems encountered in modeling a variety of transducers. The purpose of this section is three-fold: to remind the user of some requirements in generating finite element and boundary element models, to point out the most common problems encountered in implementing the coupling procedure, and to suggest ways of identifying problems when they appear.

TOOLS

There is no substitute for a graphical representation of a numerical model. The user may have carefully checked the geometry input file, but missed a detail that becomes obvious when displayed on the computer screen. This is not to imply, however, that pictures can expose all modeling errors. The model may look perfect, but the coordinates may have the wrong units, or the normals of the boundary elements may point in the wrong direction. Therefore, it is important to combine the graphical capability with other techniques.

Another way of finding errors is to use computer programs that process the input and output files and look for specific types of errors. One might write a program to compute the aspect ratio of each finite element and boundary element. Another program might compute the degree of asymmetry of the mutual coupling matrix. There is no limit to the kinds of programs that can be created for error checking.

Finally, one can try to avoid errors in the first place by automating as much of the modeling process as is practical. One part of the procedure that is particularly well suited to automation is the generation of the boundary element model. Suppose that a three-dimensional finite element model has been generated. The description of any finite element model includes both nodal coordinates and element connectivities. One could write a program that finds all exposed surfaces, whether they are interior or exterior, and generates files containing the coordinates and element connectivities of boundary elements describing these surfaces. Then the boundary element model is completely consistent with the finite element model. It should be noted that the reason this is possible is that the CHIEF program has the capability of using

geometry input supplied by external data files. There are certainly other parts of the coupling procedure that are amenable to automation.

ERRORS

Aspect Ratio

Both finite element and boundary element models are subject to numerical errors if the user is not careful to keep the aspect ratios of all elements within limits. In general, it is a good idea not to exceed an aspect ratio of 3 to 5 on any element. A three-dimensional plot of the model is a good way to find elements that are too long and thin. However, there is at least one case in which it may not be obvious when there is an aspect ratio problem, and that is when the CHIEF code is used to model an axisymmetric structure. Unlike most finite element models, in which the axisymmetric model is truly two dimensional, CHIEF takes the two-dimensional input from the user, and transforms it into a three-dimensional model. The number of elements generated around the circumference is specified by the user in the parameter nblks. The user must specify enough circumferential elements to avoid aspect ratio problems between either of the two dimensions of the input geometry and the third dimension (circumference). A new program called VIEWCHIEF displays the three-dimensional CHIEF model and is now available to government contractors along with CHIEF and XCID.

Interior Points

The necessity of using interior points in a CHIEF model was discussed in detail in the discussion on boundary elements (see equation (26)). Interior points are needed at particular frequencies corresponding to the interior resonances. For complex models, it is impossible to compute directly the frequencies at which interior points are required, but one can begin by omitting interior points. Then there are two ways to check the condition of the matrix. First, the latest release of the CHIEF code computes the condition number of the matrix. Alternatively, one can compute the $[Z]$ matrix and check the degree of asymmetry in the matrix. This is done by either computing the percent asymmetry of each component z_{ij} relative to the largest component at each frequency, or by simply drawing a surface plot of the real and imaginary parts of the matrix versus row and column numbers. This will be demonstrated in the example problem in section 7.

Boundary Element Model Not Closed

In generating the BE model, it is possible to make a mistake in the coordinates or connectivities. These errors often result in a "hole" in the wetted surface. This problem can be found by checking the symmetry of the $[Z]$ matrix.

Compatibility of FE and BE Models

There are a number of ways to make a mistake in combining a finite element model with a boundary element model. Many of them can be avoided using diagnostic and automatic generation programs, as mentioned above. Some specific items to look for are

1. Consistent coordinate systems,

2. Consistent units,

3. Consistent symmetries (CHIEF models can have planar symmetry or axisymmetry, but not both while many FE programs can combine the two),

4. Correct connectivity (BE model may require different node ordering than FE model),

5. Correct definition of surface normals (defined by connectivity),

6. Matching order of wetted surfaces for computing $[C]$ in FE model with order in BE model ($[C]$ and $[Z]$ must be consistent),

7. Boundary elements not allowed on symmetry lines/planes,

8. For an axisymmetric problem, the $[Z]$ matrix from CHIEF must be scaled by nblks/(2π) before being combined with the matrices from an axisymmetric FE model.

Other Problems

There are many other possible errors, all of which require care on the part of the user to avoid. Two items of particular importance, which have not been mentioned before, are the definition of the material polarization direction and the electrical and mechanical boundary conditions.

7. DEMONSTRATION OF THE COUPLING PROCEDURE

PROBLEM DESCRIPTION

The problem chosen for this example is a piezoelectric ceramic rod of circular cross-section (see figure 9). The rod has electrodes at either end, and is polarized along its longitudinal axis. This example will be exercised for most of the analyses presented in the preceding sections,

and the associated matrices and input/output files will be listed in the appendices. Appendix B contains the set of material constants for the piezoelectric ceramic, in this case, Navy Type I.[8] Appendix C lists the contents of the various input/output files. These contents include geometry information, finite element and boundary element matrices, and computed velocities and pressures. Appendix D contains listings of the Fortran programs needed for the analyses, including the CHIEF driver programs and the external program for computing the coupled solution. The latter program performs an *in vacuo* and in-fluid modal analysis, an *in vacuo* and in-fluid harmonic analysis with an electrical potential excitation, and an in-fluid harmonic analysis for an incident plane wave. These analyses are performed using real stiffness matrices; that is, the material is assumed to be lossless.

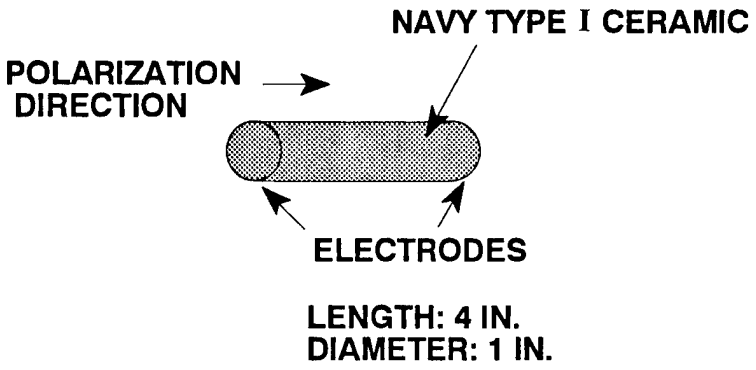


Figure 9. Circular Rod Made of Navy Type I Ceramic

Given that the rod has a circular cross-section, one is immediately tempted to use a two-dimensional (axisymmetric) model to describe the three-dimensional structure. However, for this to be appropriate, not only must the geometry be axisymmetric, but the excitation must be axisymmetric as well. For the case of an applied potential across the electrodes, this requirement is satisfied because our electrodes are axisymmetric. If the electrodes consisted of strips down the side of the rod, for example, this would not be true. For the case of mechanical excitation, the applied force must be axisymmetric, perhaps striking one end of the rod. For acoustic excitation, the plane wave must cause incident surface pressures that do not vary with angle. The only way for this to be satisfied is for the plane wave to be incident parallel to the axis of the rod, that is, for it to impinge upon either end of the rod. To keep the size of the problem within reasonable limits, all excitations in this example will be axisymmetric. This is a good choice for a tutorial for another reason: the coupling procedure is slightly more complicated for axisymmetric models than for three-dimensional models, at least when a CHIEF boundary element code is used.

The finite element model of the rod is shown in figure 10. The model includes only two elements, so the accuracy of the results is not likely to be good, except at low frequencies, but the idea is to keep the problem small. The longitudinal wavelength in the material can be computed

to determine how many elements there are per wavelength. The longitudinal wavelength in the bar at the first resonance is found by setting the wavelength equal to twice the length of the bar, or $\lambda = 2\ \ell$, where ℓ = 4.0 in. = 0.1016 m. Then the wavelength at this frequency is 0.2032 m. The length of the longest element is equal to half the length of the bar, or 0.0508 m. Then the lowest ratio of elements to structural wavelength is 4 [(0.2032 m/wavelength)/(0.0508 m/element) = 4 elements/wavelength], which is the minimum acceptable value. Therefore, we would expect a fairly accurate computation of the fundamental resonance frequency, but not as good for higher order modes.

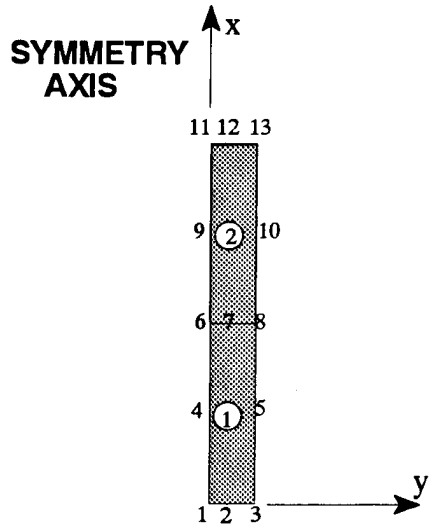


Figure 10. Axisymmetric FE Model (Element numbers are circled, node numbers are plain)

The FE model was generated with the ATILA FE code, which requires that the symmetry axis be in the x-direction, which is also the direction of polarization for this problem. The electrodes are assumed to be thin, and are not included in the model structure. The constant potential across the electrode on one end and the zero potential on the electrode at the other end are prescribed using boundary conditions. An additional boundary condition is needed to restrict motion across the axis of symmetry. The element numbers and node numbers used in the model are shown in figure 10. The element connectivities are given in table 2, and the coordinates of the nodes are given in table 3. Note that the connectivities of the quadrilateral finite elements are ordered as: left bottom corner, right bottom corner, left top corner, right top corner, followed by the mid-side nodes. This is the convention used in ATILA. The usual convention is to number counterclockwise The FE model will be used to generate the stiffness, mass, and compatibility matrices.

Table 2. FE Connectivities

Element No.	Connectivity
1	1,3,6,8,2,4,5,7
2	6,8,11,13,7,9,10,12

Table 3. FE Nodes

Node Number	Coordinates (m)		
	x	y	z
1	0.0	0.0	0.0
2	0.0	0.00635	0.0
3	0.0	0.0127	0.0
4	0.0254	0.0	0.0
5	0.0254	0.0127	0.0
6	0.0508	0.0	0.0
7	0.0508	0.00635	0.0
8	0.0508	0.0127	0.0
9	0.0762	0.0	0.0
10	0.0762	0.0127	0.0
11	0.1016	0.0	0.0
12	0.1016	0.00635	0.0
13	0.1016	0.0127	0.0

The input geometry for the boundary element model shown in figure 11, was generated automatically from the FE geometry using an automated command program developed at USRD. With this approach, there is far less possibility of the two models being incompatible than if one model was generated using the ATILA preprocessor and the other using the CHIEF preprocessor. Note that the BE model does not use all of the nodes, but the node numbers remain the same as in the FE model. The boundary elements are one-dimensional (lines), because they represent the edges of the two-dimensional finite element model. The nodal coordinates and element connectivities of the boundary elements are given in tables 4 and 5, respectively. The

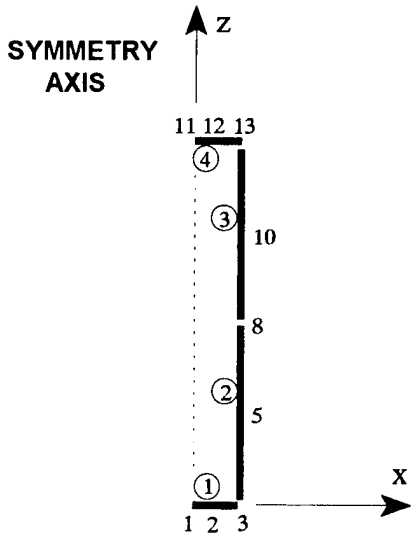


Figure 11. Two-Dimensional BE Model Input

CHIEF program uses the two-dimensional input geometry to create a three-dimensional model having nblks rotational symmetry blocks. In this case, nblks = 8. The resulting three-dimensional CHIEF model is shown in figure 12. Instead of generating the BE model from the FE geometry, the BE model for this simple geometry could have been generated directly by CHIEF,[19] using cylindrical for the sides and circular planar surfaces for the ends. This would not be an option for more complex surface geometries.

One can compute the number of boundary elements per wavelength in the fluid using the equation: $\lambda_f = c/f$, where c is the speed of sound in the fluid and f is the frequency of operation. One will see later that the *in vacuo* resonance frequency is near 15 kHz, so this can be used to get a conservative estimate of the number of elements per wavelength. The in-water frequency range of interest will be lower than this frequency. The speed of sound in seawater is approximately 1500 m/s so λ_f = (1500 m/s)/(15000 Hz) = 0.1 m. The longest boundary element is $\ell/2$ = 0.0508 m in length, so we have (0.1 m/λ)/(0.0508 m/element) ≈ 2 elements per wavelength. At 15 kHz one needs at least twice as many elements as one has, so accurate results for the BE model cannot be expected. However, in the interest of keeping the model small, the resulting numerical error will be accepted.

Table 4. BE Connectivities

Element No.	Connectivity
1	1,3,2
2	3,8,5
3	8,13,10
4	13,11,12

Table 5. BE Nodes

Node No.	Coordinates (m)	
	x	z
1	0.0	0.0
2	0.00635	0.0
3	0.0127	0.0
4	0.0	0.0254
5	0.0127	0.0254
6	0.0	0.0508
7	0.00635	0.0508
8	0.0127	0.0508
9	0.0	0.0762
10	0.0127	0.0762
11	0.0	0.1016
12	0.00635	0.1016
13	0.0127	0.1016

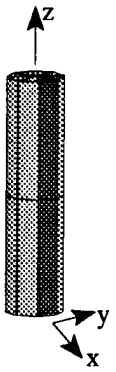


Figure 12. Three-Dimensional BE Model (First rotational symmetry block is shown in dark gray.)

COUPLING PROCEDURE RESULTS

The results of the coupling procedure are grouped in three categories as follows:

1. Components	DOF table Compatibility matrix Mutual coupling matrix
2. *In vacuo* analysis:	Modal (short- and open-circuit) Harmonic (electrical excitation)
3. In-water analysis:	Modal (short- and open-circuit) Harmonic: radiation (electrical excitation) scattering (plane wave excitation)

Components

The complete stiffness and mass matrices computed by the ATILA code are found in appendix C and will not be discussed in detail here. The degree of freedom correspondence table is presented in table 6. The asterisks indicate a restricted motion, in this case, the nodes on the axis of symmetry cannot move laterally because this would violate the assumption of axisymmetric

vibration. In the next few sections results are presented for the normal displacement at the top edge of the bar. The node number at this position is 13. Looking at table 6, one sees that the normal displacement (u_x) at node 13 is represented by DOF 20. Therefore, in the computed vector of nodal displacements, one will be interested in the 20th component.

The complete compatibility matrix [C] is shown in table 7. Note that the absolute values of the nonzero elements range from 0.16667 (rounded to 0.17) to 0.66667 (rounded to 0.67). Examine the first column of [C], which corresponds to the consistent forces on boundary element 1 (see figure 11), for an applied pressure of -1 Pa. These forces are computed by the finite element code and are determined by the particular shape function for the type of element used. In this case, an axisymmetric quadrilateral element was used, for which the consistent forces represent the combined effect of the force acting along the entire circumference.[20] For a degree of freedom at a node on the symmetry axis, the circumference at that node is $2\pi r$, where $r = 0$, so the consistent force for any degree of freedeom on the axis is zero. Looking at table 6, one sees that DOF 1 represents the x displacement at node 1, on the symmetry axis. Looking then at table 7, the consistent force pertaining to DOF 1 is the (1,1) component of the compatibility matrix, and this component is zero. Continuing down the first column in table 7, one sees that a unit pressure applied on surface 1 results in a nonzero force only in DOF 2 and 4. These correspond to the x displacements at nodes 2 and 3, respectively. No other degrees of freedom are affected by a pressure on surface 1. Note that the consistent forces in DOFs 2 and 4 are negative, because a negative unit pressure on surface 1 results in forces acting in the negative x direction. Note also that the forces sum to -1. Recall that the columns of the compatibility matrix must sum to +/-1 because the elements of the matrix represent the fraction of the total force applied in each degree of freedom, and the fractions must add up to the whole or one.

Table 6. DOF Correspondence

Node	DOF (u_x)	DOF (u_y)
1	1	*
2	2	3
3	4	5
4	6	*
5	7	8
6	9	*
7	10	11
8	12	13
9	14	*
10	15	16
11	17	*
12	18	19
13	20	21

Table 7. Compatibility Matrix

BE→ DOF↓	1	2	3	4
1	0	0	0	0
2	-0.67	0	0	0
3	0	0	0	0
4	-0.33	0	0	0
5	0	0.17	0	0
6	0	0	0	0
7	0	0	0	0
8	0	0.67	0	0
9	0	0	0	0
10	0	0	0	0
11	0	0	0	0
12	0	0	0	0
13	0	0.17	0.17	0
14	0	0	0	0
15	0	0	0	0
16	0	0	0.67	0
17	0	0	0	0
18	0	0	0	0.67
19	0	0	0	0
20	0	0	0	0.33
21	0	0	0.17	0

Rounded to two significant digits.

Continue with the second column of the compatibility matrix (table 7). For a negative unit pressure applied on surface 2, there are three forces. In this case, these forces are distributed among DOFs 5, 8, and 13, corresponding to the y displacement at nodes 3, 5, and 8, respectively. The normalized forces applied in these degrees of freedom sum to +1, because a negative unit pressure on surface 2 results in forces acting in the positive y direction.

The next component needed for the coupled solution is the mutual coupling matrix computed by CHIEF. CHIEF generates a three-dimensional model from the two-dimensional output, and computes the [Z] matrix for the three-dimensional model. To combine this matrix with the [K], [M], and [C] matrices computed from the two-dimensional FE model, one has to manipulate the [Z] matrix to obtain the mutual coupling values per radian. The three-dimensional CHIEF model was generated with eight rotational symmetry block (nblks = 8), each having a circumferential length of $2\pi r/8$. Because each component of [Z] represents the total force on a surface of this length for a unit velocity on a surface, one can compute the force per radian by multiplying each component by nblks and dividing by (2π). That is, $[Z_{axi}] = [Z_{3\text{-}D}]$*nblks/$(2\pi)$. Note that this manipulation is not needed if the FE model is three dimensional. The mutual coupling matrix of the bar computed at 14,505 Hz is as follows:

$$\mathrm{Re}\left\{\left[Z_{axi}\right]\right\} = \begin{bmatrix} 30.1 & 16.9 & -33.8 & 6.1 \\ 33.0 & 654.6 & 41.8 & -21.2 \\ -21.2 & 41.8 & 654.6 & 33.0 \\ 6.1 & -33.8 & 16.9 & 30.0 \end{bmatrix}$$

and

$$\mathrm{Im}\left\{\left[Z_{axi}\right]\right\} = \begin{bmatrix} 84.2 & -42.0 & 32.7 & -0.2 \\ -38.2 & 630.1 & -218.1 & 18.5 \\ 18.5 & -218.0 & 630.1 & -38.2 \\ -0.2 & 32.7 & -42.0 & 84.2 \end{bmatrix}$$

The largest asymmetry of the real part, computed relative to the maximum real component, is 2.5 percent, while that of the imaginary part relative to the maximum imaginary component is 2.3 percent. This amount of asymmetry is larger than desired, but it is expected because there are not enough elements in the BE model at this frequency. It is important to remember that the symmetry, or lack thereof, is relative to the scale of the matrix, which is why the percent asymmetry was computed relative to the maximum value.

The mutual coupling matrix shows that the first and fourth diagonal elements are the same. These numbers represent self impedance of the two ends of the rod. Because the two ends are identical, their impedances must also be the same. The same follows for the second and third diagonal elements, which represent the self impedances of the two elements on the side of the rod. Each of the off-diagonal terms represents the mutual coupling between two different elements.

In Vacuo Results

Modal

The results for the short- and open-circuit eigenfrequencies of the first longitudinal (fundamental) mode of the free-free bar were obtained by reading the stiffness and mass matrices computed by ATILA into an external FORTRAN program that uses LAPACK subroutines[21] and solving the appropriate eigenvalue equations. The results for the first four modes, including

short- and open-circuit eigenvalues and short-circuit eigenvectors, are given in appendix C. The program listing is given in appendix D. The results for the fundamental mode are as follows:

f_{sc} = 15,086 Hz

f_{oc} = 20,069 Hz,

and

k = 65.95 percent.

Compare these results to those obtained from an analytic bar model. The analytic equations for the short-and open-circuit frequencies of an end-electroded thin bar with parallel electric field are:[22]

$$f_{sc} = \frac{1}{2\ell}\sqrt{\frac{1}{\rho s_{33}^{E}}}\ , \tag{48}$$

and

$$f_{oc} = \frac{1}{2\ell}\sqrt{\frac{1}{\rho s_{33}^{D}}}\ , \tag{49}$$

where ℓ is the length of the bar, ρ is the density of the material, and s_{33}^{E} and s_{33}^{D} are the compliance constants under constant electric field (short circuit) and constant electric displacement (open circuit), respectively. The pertinent material properties of Navy Type I[8] are as follows: ρ = 7500 kg/m^3, $s_{33}^{E} = 15.5x10^{-12} m^2 / N$, and $s_{33}^{D} = 7.9x10^{-12} m^2 / N$. Using these values in equations (48), (49), and (24), respectively, one obtains:

f_{sc} = 14433.8 Hz,

f_{oc} = 20217.7 Hz,

and

k = 70.0 percent.

These frequencies are within a few percent of the finite element results. A perfect match is not expected because the analytic model assumes a thin bar (no lateral stress).

The modal results give information about the behavior of the bar in each mode and tell us the frequencies at which the harmonic response will peak for the same boundary conditions. Because the bar is free at both ends, the eigenfrequencies relate to the harmonic peaks for either a force or potential excitation. In particular, the short-circuit eigenfrequencies relate to the harmonic peaks if the piezoelectric material is used as a driver, equation (19). On the other hand, the open-circuit eigenfrequencies give us the harmonic peaks when the material is used as a sensor, equation (18).

If there is interest in applying a constant displacement excitation, rather than a constant force or voltage in the harmonic analysis, the corresponding eigenfrequencies are found by applying a rigid boundary condition at one end of the bar.

Harmonic

The displacements of the bar were computed for an applied electrical potential of 1 V, at frequencies near the *in vacuo* eigenfrequency (15,086 Hz). Figure 13 shows the displacement of the center of one end of the bar (node 13, DOF 20) versus frequency. The resonance peak is very sharp because of the absence of material losses in the model, and occurs at exactly the short-circuit eigenfrequency 15,086 Hz. The full set of displacements for all degrees of freedom at frequencies between 15,080 Hz and 15, 090 Hz is given in appendix C.

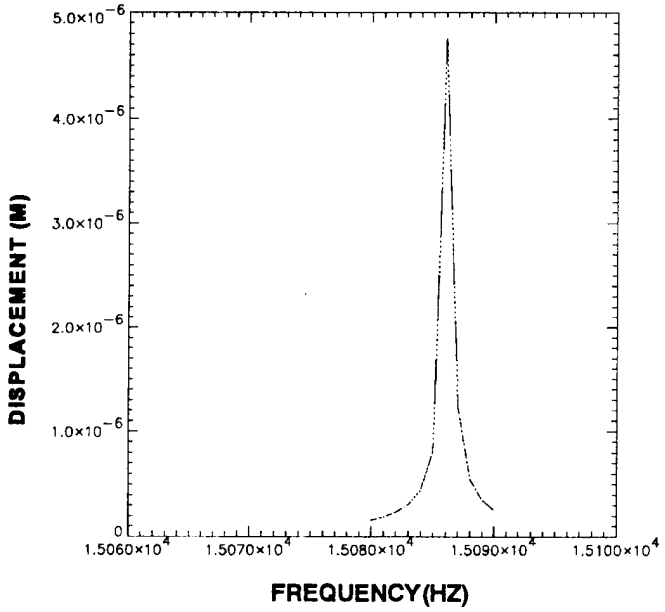


Figure 13. In Vacuo X-Displacement at Node 13 for 1-V Input

In-Water Results

Modal

The in-fluid modal analysis was performed for the bar in sea water (ρ = 1000 kg/m^3, c = 1500 m/s), under both short- and open-circuit conditions. For the short-circuit case, beginning with the mutual coupling matrix computed at the *in vacuo* eigenfrequency, 15,086 Hz, the procedure performs successive eigenvalue computations until the in-fluid eigenfrequency converges for the mode of interest. At the beginning of the program execution, the user is asked for the *in vacuo* eigenfrequency. The program then runs CHIEF to obtain the mutual coupling matrix at this frequency, modifies the mass matrix using the imaginary part of [*Z*], and computes the eigenfrequencies of the modified system. At this point, the user is offered a choice of modes. For the current problem, the fundamental mode is the second eigenfrequency, both *in vacuo* and in water. The first eigenfrequency corresponds to a rigid body mode.

After the first computation, the user chooses from the following list of modes:

- Mode 2 - 14,542.9,
- Mode 3 - 41,357.4,
- Mode 4 - 61,115.3.

These are the frequencies of the second, third, and fourth modes computed, using the [*Z*] matrix obtained at 15,086 Hz. In this case, one knows that the second mode corresponds to the mode of interest, so choose mode 2. For other problems in which the in-fluid modes may be close together in frequency, it is possible for the order of the modes in water to be different from the order *in vacuo*. For this reason, the user is cautioned to check the mode shapes after the first computation, using a graphical interface or other method, to be sure the correct mode is chosen.

Once mode 2 is specified, the program proceeds to compute the [*Z*] matrix at 14,542.9 Hz, and then computes the new set of eigenfrequencies, using the new modified mass matrix. This procedure continues until the frequency converges for the mode of interest. The succession of eigenfrequencies for mode 2, including the *in vacuo* frequency, is as follows:

- 15,086.0 Hz,
- 14,542.9 Hz,
- 14,527.4 Hz, and
- 14,527.0 Hz.

Thus, the in-water eigenfrequency of the fundamental mode under short-circuit conditions is estimated to be 14,527.0 Hz. The eigenvector, or mode shape, is given in appendix C.

The progression of in-water open-circuit eigenfrequencies for mode 2 is as follows:

- 20,069.0 Hz,
- 19,554.84 Hz, and
- 19,555.78 Hz.

Then, the in-water eigenfrequency of mode 2 under open-circuit conditions is approximately 19,556 Hz.

Harmonic Radiation

The displacement of the bar in sea water, for a 1-V input (driver), is shown in figure 14 for frequencies between 14,500 Hz and 14,530 Hz. The peak in the harmonic response is at 14,513 Hz. Recall that the in-water modal procedure gave 14,527 Hz, which is a difference of 0.1 percent. The complete set of displacements, as well as normal velocities, is given in appendix C. If one is interested in the radiated pressures, one can provide the normal velocities to the CHIEF driver program listed in appendix D. The resulting farfield pressures are given in appendix C.

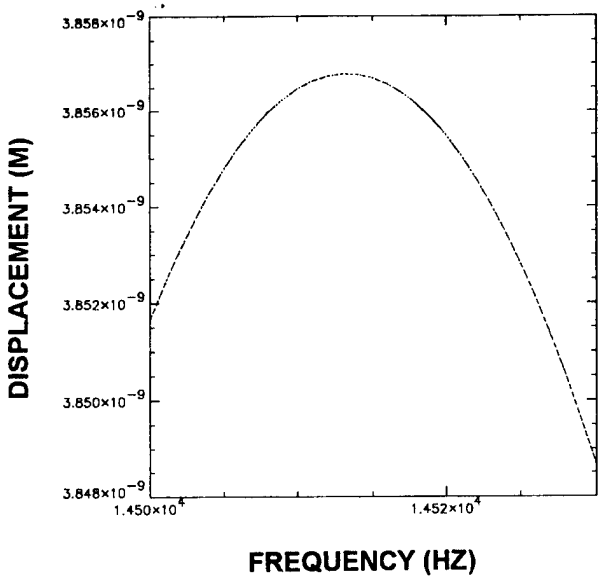


Figure 14. In-Water X-Displacement at Node 13 for 1-V Input

Harmonic Scattering

This displacement of the bar excited by an incident plane wave of unit amplitude is shown in figure 15 for frequencies between 19, 350 Hz and 19, 380 Hz. In this case, the bar acts as a sensor, and the open-circuit stiffness matrix, equation (17), is used. The peak in the harmonic response occurs at 19,374 Hz, compared to the in-water modal estimate of 19,555 Hz, which represents a difference of 0.9 percent. The complete displacements, normal velocities, and scattered pressures are given in appendix C, and the CHIEF driver programs are listed in appendix D.

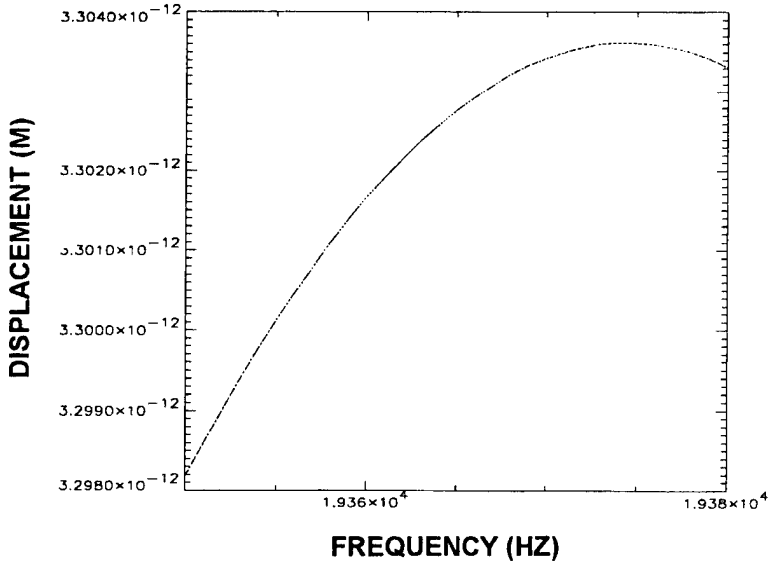


Figure 15. In-Water X-Displacement at Node 13 for Incident Plane Wave

8. SUMMARY

It is hoped that this document will help interested Navy personnel and contractors to understand the components and procedures involved in the numerical analysis of fluid-loaded structures, and to use them to solve problems of practical significance. The authors have attempted to provide enough detail to explain the subject, without burdening the reader with too many derivations. A more in-depth treatment of the various numerical techniques may be found in references 1, 2, and 3.

After reading the first three sections of this document, the user should be able to duplicate the results provided for the example problem, assuming that the necessary tools are available. As stated in the introduction, another possible use of this document is to provide the relevant equations to the authors/vendors of a finite element code. Then, either the required matrices can be written to external files, or the mutual coupling matrix can be brought into the finite element code and the equations solved internally. A boundary element code is required in any case; the CHIEF code is available to Department of Defense installations and their contractors (for the duration of the contract).[7]

APPENDIX A

THEORETICAL DEVELOPMENT OF THE FLUID-STRUCTURE COUPLING APPROXIMATION

Consider a bounded finite radiating structure surrounded by an infinite fluid. The farfield pressure can be found if the normal velocity or pressure is known for each patch of surface. Each patch is assumed to be a face of some element within the structure. Also, assume that the normal velocity varies slowly within a given patch. The structure may be composed of a variety of solid materials, at least one of which is piezoelectrically active. Passive structures with known boundary conditions can also be included, either as an integral part of the radiator or as a separate structure in the fluid. Two common examples are the rigid baffle and the pressure release surface. Assume that a sinusoidal voltage is applied to the electrodes of the active component so that the structure is driven into linear steady-state vibration with time dependence $\exp(j\omega t)$ where ω is the frequency in radians per second. For sufficiently small vibrations, the dynamics of the structure are governed by Maxwell's Equation, Hooke's Law, and the continuum equivalent of Newton's Second Law. Similarly, the classical wave equation describes the behavior of the fluid. A detailed description of these equations is not central to the theme of this document, but the interested reader may refer to references 22 through 25. These equations can be solved analytically when simplifying approximations can be made on the basis of geometrical or dynamical considerations; however, the solutions are valid only in a limited frequency range. Analytical solutions, even when available, often fail to predict much of the fine structure in the modal response of "real" systems of rather complex geometry and boundary conditions.

An alternate approach is to cast the dynamical equations in integral form (the "weak" formulation). Then, the dependent variables can be expressed as an expansion in terms of a finite set of basis functions defined within discrete domains that span the structure and fluid. This so-called "finite-element" discretization converts the continuum problem into a matrix equation that can be solved with numerical algorithms and digital computers. The application of this approach to piezoelectric structures is documented in references 1, 12, and 26 through 29. The surrounding fluid can also be discretized in this way; however, when one boundary is at infinity, approximations must be made that limit the range of validity. Also, in many instances, the resultant matrix equations are of such large dimension that practical solution is not possible. Alternatively, the acoustic field within the fluid can be represented by the Helmholtz Integral Equation. This approach was the basis for the numerical code CHIEF developed by Schenck[2] and Benthien.[5] With this method, the pressure and velocity, at any point in the fluid, can be found provided that the pressure or velocity is known everywhere on the surface of the embedded structure. CHIEF solves the acoustic problem but not the structure problem. The structure problem can be solved by FE methods provided that the interaction with the fluid is correctly included. Therefore, the complete structure/fluid problem requires a method for coupling an FE description of the structure to a CHIEF-type formulation of the acoustic problems. It is important to recognize that the structure and acoustic problems cannot be solved independently, but must be

solved as a coupled system. A coupling approach is presented here that has been successfully applied to a number of transducer problems. Similar approaches have been used by others to solve acoustic scattering problems[5] and noise fields arising from vibrating structures.[30]

The FE discretization of the equations of motion for the radiator leads to a matrix equation of the form:

$$\left[\begin{pmatrix} K_{UU} & K_{U\Phi} \\ K_{U\Phi}^{t} & K_{\Phi\Phi} \end{pmatrix} - \Omega^2 \begin{pmatrix} M & O \\ O & O \end{pmatrix}\right] \begin{Bmatrix} \mathbf{u} \\ \boldsymbol{\phi} \end{Bmatrix} = \begin{Bmatrix} \mathbf{f} \\ \mathbf{q} \end{Bmatrix} \tag{A-1}$$

where $[K]$ refers to stiffness, $[M]$ to mass, $\{\mathbf{u}\}$ the generalized displacement vector, $\{\mathbf{f}\}$ the consistent applied force vector, $\{\boldsymbol{\phi}\}$ the electrical potential, $\{\mathbf{q}\}$ the charge on the electrodes, and Ω the radian excitation frequency. Equation (A-1) has been written in a partitioned form that separates the electrical degrees of freedom from the spatial degrees of freedom. In general, $\{\boldsymbol{\phi}\}$ is a vector that includes potentials on the electrodes, as well as points interior to the piezoelectric material. However, if the interior is a good insulator ($\{\mathbf{q}\} = 0$), then the interior potentials can be expressed in terms of $\{\mathbf{u}\}$ and the potentials on the electrodes and thus are eliminated from equation (A-1). In doing so, one forms the short-circuit stiffness matrix, $[K'_{UU}]$, from the elastic stiffness matrix, $[K_{UU}]$. Furthermore, if one assumes that the electrodes are equipotential surfaces, and one of these is ground, then the matrix of potentials, $\{\boldsymbol{\phi}\}$, reduces to a scalar quantity, ϕ_E. Therefore, our analysis assumes that $\{\boldsymbol{\phi}\}$ and $\{\mathbf{q}\}$ can be reduced to scalars. Then the matrix $[K_{U\phi}]$ is replaced by the vector $\{\mathbf{K_{UE}}\}$, and $[K_{\phi\phi}]$ is replaced by K_{EE}. $[K_{UU}]$ depends on the elastic constants, $\{\mathbf{K_{UE}}\}$ depends on the piezoelectric constants, and K_{EE} (which is essentially the clamped capacitance of the piezoelectric driver) depends on the dielectric permittivities.

Discretization of the Helmholtz Integral Equation on the boundary of the radiating structure leads to a matrix equation of the form

$$[A]\{\mathbf{p}\} = [B]\{\mathbf{v_N}\}, \tag{A-2}$$

where A and B are square matrices that are functions of frequency, obtained by evaluating integrals in the Helmholtz Integral Equation over the discretized surface of the radiator, $\{\mathbf{p}\}$ and $\{\mathbf{v_N}\}$ are vectors of pressure and normal velocity at discrete points on the surface. In its simplest formulation, CHIEF assumes that the surface can be subdivided into regions wherein the pressure and velocity are constant with position. Thus, the entire surface can be visualized as being made up of a number of contiguous vibrating rigid pistons of various shapes and sizes.

Similarly, discretized equations can be derived for the near- and farfield pressures in terms of the surface values $\{\mathbf{p}\}$ and $\{\mathbf{v_N}\}$. The essential point here is that equations (A-1) and (A-2) constitute a complete description of the problem, but they must be solved simultaneously. To do

this, $\{\mathbf{p}\}$ may be converted to a force vector by multiplying each of its components by the area of the corresponding subdivision. Then equation (A-2) can be rearranged as

$$\{\mathbf{f}_N\} = [Z]\{\mathbf{v}_N\} , \tag{A-3}$$

where each component of $\{\mathbf{f}_N\}$ is the average normal force action on a given subdivision and $[Z]$ is the radiation impedance matrix. This force not only incorporates the radiation and mass loading terms but also includes the interelement acoustic coupling among subdivisions. Assume that the spatial discretization chosen for the FE formulation meshes with the subdivisions chosen in equation (A-2). That is, each CHIEF subdivision is a face of some finite element. Then, the normal displacements $\{\mathbf{u}_{N_i}\}$ at any point on subdivision i can be expressed in terms of the nodal displacements as

$$\{\mathbf{u}_{N_i}\} = \{\mathbf{n}_i\}^t [N] \{\mathbf{u}\}, \tag{A-4}$$

where $\{\mathbf{n}_i\}^t$ is a unit normal vector, $[N]$ is a matrix of interpolation functions in the global coordinate system [1], and $\{\mathbf{u}\}$ is a vector of nodal displacements. The average normal displacement can then be found by averaging the vector $\{\mathbf{u}_{N_i}\}$ over each patch of the fluid-to-structure interface. Thus, for subdivision i,

$$\mathbf{u}_{N_{av_i}} = \frac{1}{S_i} \int\int \{\mathbf{n}_i\}^t [N] \{\mathbf{u}\} \, dS_i , \tag{A-5}$$

where S_i is the area of subdivision i. Equation (A-5) thus provides the components of a vector of normal displacements, $\{\mathbf{u}_{Nav}\}$, that, for harmonic motion, are related to the centroid velocities in equation (A-3) as

$$\{\mathbf{v}_N\} = j\Omega \{\mathbf{u}_{N_{av}}\}. \tag{A-6}$$

Here $\{\mathbf{u}_{Nav}\}$, a vector of the same dimension as $\{\mathbf{v}_N\}$, has been assembled from the components obtained from equation (A-5) for each subdivision of the interface. Combining equations (A-3), (A-5), and (A-6), the pressure on subdivision i can be expressed as

$$p_i = \frac{j\Omega}{S_i} \sum_k \frac{z_{ik}}{S_k} \int\int \{\mathbf{n}_k\}^t [N] \{\mathbf{u}\} \, dS_k , \tag{A-7}$$

where the sum extends over all subdivisions on the interface. The contributions p_i from all surfaces can be assembled to form the vector $\{\mathbf{p}\}$ of equation (A-2), of dimension equal to the number of surface subdivisions. Returning to the FE formulation, the consistent force vector resulting from a pressure applied to the interface is, by definition,

$$\{\mathbf{f}\} = -\sum_i \int\int p_i \, [N]^t \, \{\mathbf{n_i}\} \, dS_i. \tag{A-8}$$

Substituting equation (A-7) into (A-8), and noting that $\mathbf{u}$ is not a function of the variables of integration, one obtains

$$\{\mathbf{f}\} = -j\Omega \left[\sum_i \int\int_{S_i} \frac{[N]^t \, \{\mathbf{n_i}\}}{S_i} \, dS_i \sum_k z_{ik} \int\int_{S_k} \frac{\{\mathbf{n_k}\}^t \, [N]}{S_k} \, dS_k \right] \{\mathbf{u}\}. \tag{A-9}$$

Note that either integrand in equation (A-9) is the transpose of the other. Also note from equation (A-8) that, for each summation index, these integrals can be generated as the consistent force vector produced by a pressure of -1.0 N/m^2 applied to the interface subdivision corresponding to that index. Thus, the integrations can be performed by the FE code by applying negative unit pressures to each subdivision in turn.

Suppose there are nsurf subdivisions on the surface of the structure. Let ndof denote the number of dynamic degrees of freedom required for the element discretization. The consistent load (or force) "formation" and "assembly" functions of a FE code can generate an ndof by nsurf matrix $[C]$, having the following structure: each column j of C is the consistent load vector obtained by applying negative unit pressure to subdivision j and zero pressure to all other subdivisions. Thus, the $[C]$ matrix appears as

$$[C] = \overset{\uparrow}{\underset{\downarrow}{ndof}} \left[\overset{\leftarrow nsurf \rightarrow}{(p_1 = -1.0)(p_2 = -1.0) \; .... \; (p_{nsurf} = -1.0)} \right], \tag{A-10}$$

where each column is a consistent load vector. $[C]$ is referred to as the "compatibility matrix" because it provides a way of relating the CHIEF variables to the nodal variables of the FE model.

Now one can easily show that equation (A-9) can be written as

$$\{\mathbf{f}\} = -j\Omega \, [C] \, [Z] \, [C]^t \, \{\mathbf{u}\}. \tag{A-11}$$

In addition, equations (A-4), (A-5), and (A-6) can be combined as

$$\{\mathbf{v_N}\} = j\Omega\ [C]^t\ \{\mathbf{u}\}. \qquad \text{(A-12)}$$

Equations (A-11) and (A-12) provide the relationships needed to couple the structure problem to the acoustic radiation problem. These equations are approximations in the sense that nodal degrees of freedom are averaged over the element faces to obtain centroid values that are compatible with the CHIEF code.

Now the structure problem, including fluid loading, can be solved explicitly, as follows. From the first row of equation (A-1), and recalling that ϕ_E relates to the external potential, one finds

$$[K'_{UU} - \Omega^2 M]\ \{\mathbf{u}\} + [K_{UE}]\,\phi_E = \{\mathrm{f}\}. \qquad \text{(A-13)}$$

Using the coupling relations, one can write equation (A-13) as

$$[K'_{UU} - \Omega^2 M]\ \{\mathbf{u}\} + [C]\ [Z]\ \{\mathbf{v_N}\} = -\{\mathbf{K_{UE}}\}\,\phi_E\ , \qquad \text{(A-14)}$$

or

$$\{\mathbf{u}\} + [K'_{UU} - \Omega^2 M]^{-1}\ [C]\ [Z]\ \{\mathbf{v_N}\} = -\ [K'_{UU} - \Omega^2 M]^{-1}\ \{\mathbf{K_{UE}}\}\,\phi_E. \qquad \text{(A-15)}$$

Multiplying both sides by $j\Omega[C]^t$ and combining terms, one finds

$$\left[I + j\Omega[C]^t\ [K'_{UU} - \Omega^2 M]^{-1}\ [C]\ [Z]\right]\{\mathbf{v_N}\} = j\Omega[C]^t\ [K'_{UU} - \Omega^2 M]^{-1}\ \{\mathbf{K_{UE}}\}\,\phi_E. \qquad \text{(A-16)}$$

Every matrix in this equation except [Z] can be generated by the FE code. The impedance [Z] can be generated by CHIEF. Equation (A-16) can be solved for $\{\mathbf{v_N}\}$ by using standard numerical solvers. The results can then be entered into equation (A-2), which can then be solved to yield surface pressures and subsequently field pressures at any point in the fluid. Alternative formulations can also be used; for example, one could set up the problem in terms of $\{\mathbf{u}\}$ rather than $\{\mathbf{v_N}\}$, as done in the equations in the main body of this document. However, note that the dimension of $\{\mathbf{u}\}$ (nodal displacements) is always much larger than $\{\mathbf{v_N}\}$ (surface normal velocities) except for rare pathological cases. Thus, equation (A-16) is a more efficient formulation than one involving $\{\mathbf{u}\}$.

When solutions are desired for a large number of frequencies, computational time may become prohibitive because the $[K'_{UU} - \Omega^2 M]^{-1}[C]$ term has to be evaluated for each frequency. Computation time can be reduced by diagonalizing the stiffness and mass matrices using the normal modes of the free structure. Let $[\psi]$ be a matrix of eigenvectors of the short-circuit eigenvalue problem; i.e., each column of $[\psi]$ is an eigenvector of $[K'_{UU} - \Omega^2 M]$. Then, with appropriate normalization, the stiffness and mass can be diagonalized as

$$\begin{aligned} [\psi]^t\ [K'_{UU}]\ [\psi] &= [\lambda^2], \\ [\psi]^t\ [M]\ [\psi] &= [I] \end{aligned} \qquad \text{(A-17)}$$

where $[\lambda^2]$ is a diagonal matrix, made up of the squares of the eigenvalues corresponding to the *in vacuo* normal modes. Using equation (A-17), equation (A-16) can be rewritten as

$$\left\{[I] + j\Omega\ [C]\ [\psi]^t[\lambda^2 - \Omega^2 I]^{-1}\ [\psi]\ [C]\ [Z]\right\}\ \{v_N\} = -j\Omega\ [C]^t\ [\psi][\lambda^2 - \Omega^2 I]^{-1}\ [\psi]^t\ [K_{U\Phi E}]\,\phi_E. \qquad \text{(A-18)}$$

Since $[\lambda^2 - \Omega^2 I]$ is diagonal, it can be quickly inverted for each new frequency. Although equation (A-18) is generally more computationally efficient than equation (A-16), it may be prone to numerical error when computing modal frequencies and mode shapes, especially when large numbers of degrees of freedom are involved.

APPENDIX B

PIEZOELECTRIC MATERIAL CONSTANTS

$\rho = 7500.0$ kg/m^3

$[s^E]$:

1.280E-11	-4.050E-12	-5.310E-12	0.000E+00	0.000E+00	0.000E+00
-4.050E-12	1.280E-11	-5.310E-12	0.000E+00	0.000E+00	0.000E+00
-5.310E-12	-5.310E-12	1.550E-11	0.000E+00	0.000E+00	0.000E+00
0.000E+00	0.000E+00	0.000E+00	3.900E-11	0.000E+00	0.000E+00
0.000E+00	0.000E+00	0.000E+00	0.000E+00	3.900E-11	0.000E+00
0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00	3.270E-11

$[d]$:

0.000E+00	0.000E+00	0.000E+00	0.000E+00	4.960E-10	0.000E+00
0.000E+00	0.000E+00	0.000E+00	4.960E-10	0.000E+00	0.000E+00
-1.200E-10	-1.200E-10	2.890E-10	0.000E+00	0.000E+00	0.000E+00

$[\varepsilon^S]$:

6.460E-09	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00
0.000E+00	6.460E-09	0.000E+00	0.000E+00	0.000E+00	0.000E+00
0.000E+00	0.000E+00	5.620E-09	0.000E+00	0.000E+00	0.000E+00

APPENDIX C

INPUT/OUTPUT FILES

This appendix includes the files created and used by the various Fortran programs (given in appendix D), as well as the results of the analyses described in the main text. The files are given in the following order:

- Stiffness, mass, and compatibility matrices, and degree of freedom correspondence table,
- Results of external solver program for *in vacuo* modal analysis (eigenfrequencies and eigenvectors),
- Results of external solver program for *in vacuo* harmonic analysis (nodal displacements, input admittance),
- Two-dimensional geometry input files for CHIEF,
- $[Z]$ matrix at 15,086 Hz computed in CHIEF run for in-water modal analysis,
- Results of external solver program for in-water modal analysis (eigenfrequencies and eigenvectors),
- $[Z]$ matrix at 14,513 Hz computed by CHIEF for radiation analysis,
- Results of external solver program for radiation analysis (nodal displacements, normal velocities, input admittance),
- Results of second CHIEF run for radiation analysis (far-field pressures at 14,513 Hz),
- $[Z]$ matrix at 19,374 Hz computed in first CHIEF run for scattering analysis,
- Pressures on rigid surface, $\{\mathbf{p}_{sr}\}$, at 19,374 Hz computed by first CHIEF run for scattering analysis,
- Pressures scattered by rigid surface, $\{\mathbf{p}_r\}$,at 19,374 Hz computed by first CHIEF run for scattering analysis (also target strength),
- Results of external solver program for scattering analysis (nodal displacements, normal velocities, sensor voltage),
- Results of second CHIEF run for scattering analysis (elastic scattered pressures, total scattered pressures, target strength at 19,374 Hz).

The eigenvectors for the modal analyses are given for the second, third, and fourth modes (the first mode is a rigid body mode). The results for the harmonic analyses are provided as follows: the displacement at one node versus frequency, and the complete set of displacements at the frequency of the peak in the harmonic response.

```
********************************************************************************
File: test1.matrices
Used by program: solver.f
Used for analyses: all
Contents: matrix dimensions, elastic stiffness, piezoelectric stiffness, dielectric stiffness, mass,
compatibility, DOF table
Format: free
********************************************************************************
C* MECHANICAL, INTERNAL POT, EXTERNAL POT, CHIEF SURFACES, NODES,
CPPDC WIDTH
 C* IMECA,IELIN,IELEX,NSURF,IP,IX
      21       7       1       4       13       7
 C* KUU
  342489662.0241581    -564859664.8818606     1594838620.457041
 -195816568.7532751    -37389375.08978783     8663868506.673552
  331751954.8461463    -889504478.4692582     233205973.0861885
  846753666.3385376    -38260496.24331605     177983756.9062005
 -4307134667.221545    -504065184.4129776     3561444016.287783
 -307677943.4305750     576181795.1502001     309844478.6753820
 -487267755.6965127     145881567.7131490     903493219.8113927
  88914039.45368767    -576181795.1502007    -309844502.2271719
  49739947.44067653     323582226.9847128    -465965411.8576182
  1341021028.369291    -9159737.345214099    -491862669.8275824
  2235848.206484031    -17417663.24239773     613994295.0360957
 -1110261.582086593     1110261.582086295     5507260824.136363
  289426132.5617709    -574910205.8198695    -15605872.65862008
  394866025.2465424    -142953353.8681111    -308755398.6721492
  89991494.69526182     1480031.617979805     1039896246.750048
 -572282842.3734994     1014830656.966325     185650278.2061790
 -606620742.5703070     56743899.00252202     572090371.0917776
 -572090371.0917770     599608467.0250430    -1716305855.983677
  4838275955.703214     27624094.03198393    -195602893.1135049
  4289286102.082991     167978804.7849012    -2170600309.251019
 -309951715.9555697     309951739.5073597    -92.95666797578467
-6.1457264237105846E-03  1.0730747133493423E-02   17326706188.62283
  392238661.8001721    -603993379.1239364    -170044411.2268502
  348482157.4225922    -31156488.02525724    -482098876.3965155
  44571068.14067943    -131624715.9519619     895173505.1892679
 -2793824259.807074   -4.5850793831050396E-03   2500251466.868074
  122102332.2290325    -90109593.14319061    -2170600309.438429
  60885058.12307212     734101699.2528471    -145432555.4864986
 -324031239.2113633     608246670.1318035   -6.0041770339012146E-03
 1.0483562946319580E-02 -8611438674.745224    -4.4793039560317993E-03
```

7104223835.646496	127160.5672039562	499851.4347825055
-107237.2901583360	-627012.0019864585	449012.1984703725
42040.01508824923	-42040.01508825051	-1110261.552176607
-308755392.9113209	572090400.6539202	309951715.9839140
-482098895.6769881	145432555.5141896	903493219.8089329
-127160.5672039561	-499851.4347824655	107237.2901583350
627012.0019864687	-449012.1984703512	-42040.01508824923
42040.01508825051	1110261.552176668	89991488.93443072
-572090368.5689179	-309951739.5357037	44571103.46364790
324031239.1836723	-465965411.8551527	1341021028.366836
9159737.395173341	-31144244.87087780	-2235940.789724935
-26991780.24842843	-5747622.054832421	1110261.582086593
-1110261.582086340	-98552778.03579801	-1480031.617980041
-599608467.0250427	-89.02580703049898	131624715.9519620
608246775.3872288	1110261.552176625	-1110261.552176684
5507260824.136364	23319710.44696259	-44719318.19162777
2032434.467534968	21399607.74466514	-1270618.628307851
127160.5593452975	-127160.5593452966	-9159737.359221522
289426132.5490295	-572282857.1274313	-27624094.28096848
392238668.5455964	-122102332.4722788	-307677949.1162332
88914045.13934311	9159737.309262235	342489650.7983866
-44719318.21498418	114865818.6961485	-6025746.035045121
-46548049.60960595	18800347.06558232	499851.4708957076
-499851.4708958641	31144244.85819234	-574910190.9720585
1014830657.070827	195602892.2700455	-603993378.0337499
90109592.31917030	576181765.5747961	-576181797.6597981
491862669.8402678	-564859632.5984728	1594838521.537650
-2032434.504842205	6025745.855965633	-515728.9420479616
-3993311.339078913	1415488.801386158	107237.2801876105
-107237.2801876138	-2235940.416297887	15605872.66476593
-185650278.2169099	4289286102.155987	170044411.2314351
-2170600307.775113	-309844478.6937556	309844502.2455456
2235844.649050985	195816569.0395670	37389376.11232654
8663867873.367743	21399607.76802155	-46548049.63296238
3993311.555465637	36947667.30652341	-5285656.494196605
-627012.0302410349	627012.0302409828	26991780.22516226
394866018.4327248	-606620743.9633796	-167978803.6924572
348482157.6082340	-60885057.05580575	-487267736.4779540
49739912.17961138	17417663.26566380	331751941.8097818
-889504395.7907521	-233205974.3950191	846753620.7601513
1270618.577391028	-18800346.03021314	1415488.988797317
5285657.305520914	-9332248.828105859	-449012.2266504765
449012.2266505696	-5747619.205372039	142953353.8741152

-56743899.01300566 -2170600311.146529 31156488.02973663
734101700.1986542 -145881567.7126600 -323582226.9852021
613994192.6301295 38260496.53747914 -177983757.1175492
-4307134348.897029 504065182.5343872 3561443713.669027
C* KUE
3.4694469519536142E-17
8.3773400179883439E-03
-4.2757576992902280E-12
4.1886700110544337E-03
4.3465884447311150E-03
-6.9388939039072284E-18
2.4286128663675299E-16
1.7386353761821814E-02
3.1225022567582528E-17
1.2490009027033011E-16
-8.5514876430048403E-12
6.9388939039072284E-17
8.6931775269548043E-03
0.0000000000000000E+00
-2.0816681711721685E-17
1.7386353761822054E-02
2.2551405187698492E-17
-8.3773400179883786E-03
-4.2757481583111101E-12
-4.1886700110543486E-03
4.3465878072389674E-03
C* KEE
-4.4608749640264219E-12
C* MASS
6.8279430879133671E-04 -1.3655886173063742E-03 5.4623544688347002E-03
0.0000000000000000E+00 0.0000000000000000E+00 5.4623544688347002E-03
3.4139715417057139E-04 -6.8279430806484351E-04 0.0000000000000000E+00
1.3655886190177256E-03 0.0000000000000000E+00 0.0000000000000000E+00
-6.8279430806484351E-04 0.0000000000000000E+00 1.3655886190177256E-03
-1.0241914629619082E-03 2.7311772346127481E-03 0.0000000000000000E+00
-1.0241914629619082E-03 0.0000000000000000E+00 2.7311772346127481E-03
-1.7069857716508394E-03 4.0967658519906122E-03 0.0000000000000000E+00
-1.0241914624093094E-03 0.0000000000000000E+00 2.7311772346127481E-03
8.1935317030566515E-03 0.0000000000000000E+00 0.0000000000000000E+00
4.0967658519906122E-03 0.0000000000000000E+00 -1.0241914624093094E-03
0.0000000000000000E+00 0.0000000000000000E+00 8.1935317030566515E-03
5.1209573148095400E-04 -1.3655886173063738E-03 0.0000000000000000E+00
5.1209573148095410E-04 0.0000000000000000E+00 -1.0241914629619080E-03

-1.7069857716508399E-03 0.0000000000000000E+00 1.3655886326039745E-03
-1.3655886173063738E-03 2.7311772346127476E-03 0.0000000000000000E+00
-1.3655886173063742E-03 0.0000000000000000E+00 2.7311772346127481E-03
4.0967658519906122E-03 0.0000000000000000E+00 -2.7311772746695519E-03
1.0924709338237439E-02 0.0000000000000000E+00 0.0000000000000000E+00
2.7311772346127476E-03 0.0000000000000000E+00 -1.3655886173063742E-03
0.0000000000000000E+00 0.0000000000000000E+00 4.0967658519906122E-03
0.0000000000000000E+00 0.0000000000000000E+00 1.0924709338237439E-02
5.1209573148095410E-04 -1.3655886173063738E-03 0.0000000000000000E+00
1.7069857686018879E-04 0.0000000000000000E+00 -1.0241914629619082E-03
-1.0241914624093088E-03 0.0000000000000000E+00 6.8279428330564025E-04
-1.3655885760728832E-03 0.0000000000000000E+00 2.7311773231561594E-03
0.0000000000000000E+00 0.0000000000000000E+00 -1.3655886173063738E-03
0.0000000000000000E+00 1.7069857686018879E-04 0.0000000000000000E+00
0.0000000000000000E+00 -1.0241914624093088E-03 0.0000000000000000E+00
0.0000000000000000E+00 -1.3655885760728832E-03 0.0000000000000000E+00
2.7311773231561594E-03 0.0000000000000000E+00 0.0000000000000000E+00
0.0000000000000000E+00 0.0000000000000000E+00 0.0000000000000000E+00
0.0000000000000000E+00 0.0000000000000000E+00 0.0000000000000000E+00
-1.0241914529477069E-03 2.7311773147263548E-03 0.0000000000000000E+00
-1.0241914529477069E-03 0.0000000000000000E+00 2.7311772346127481E-03
0.0000000000000000E+00 0.0000000000000000E+00 0.0000000000000000E+00
0.0000000000000000E+00 0.0000000000000000E+00 0.0000000000000000E+00
0.0000000000000000E+00 0.0000000000000000E+00 -1.7069857816650407E-03
4.0967659721610232E-03 0.0000000000000000E+00 -1.0241913722815006E-03
0.0000000000000000E+00 2.7311772346127472E-03 8.1935317030566498E-03
0.0000000000000000E+00 0.0000000000000000E+00 0.0000000000000000E+00
0.0000000000000000E+00 0.0000000000000000E+00 0.0000000000000000E+00
0.0000000000000000E+00 0.0000000000000000E+00 0.0000000000000000E+00
0.0000000000000000E+00 4.0967659721610232E-03 0.0000000000000000E+00
-1.0241913722815006E-03 0.0000000000000000E+00 0.0000000000000000E+00
8.1935317030566498E-03 0.0000000000000000E+00 0.0000000000000000E+00
0.0000000000000000E+00 0.0000000000000000E+00 0.0000000000000000E+00
0.0000000000000000E+00 0.0000000000000000E+00 0.0000000000000000E+00
5.1209573148095400E-04 -1.3655886573631776E-03 0.0000000000000000E+00
5.1209572146675316E-04 0.0000000000000000E+00 -1.0241914729761090E-03
-1.7069857616366388E-03 0.0000000000000000E+00 6.8279429377003523E-04
0.0000000000000000E+00 0.0000000000000000E+00 0.0000000000000000E+00
0.0000000000000000E+00 0.0000000000000000E+00 0.0000000000000000E+00
0.0000000000000000E+00 0.0000000000000000E+00 -1.3655885772495705E-03
2.7311772346127476E-03 0.0000000000000000E+00 -1.3655885572211683E-03
0.0000000000000000E+00 2.7311771544991405E-03 4.0967657318201995E-03
0.0000000000000000E+00 -1.3655885772495705E-03 5.4623540682666614E-03

0.0000000000000000E+00 0.0000000000000000E+00 0.0000000000000000E+00
0.0000000000000000E+00 0.0000000000000000E+00 0.0000000000000000E+00
0.0000000000000000E+00 0.0000000000000000E+00 0.0000000000000000E+00
0.0000000000000000E+00 2.7311772346127476E-03 0.0000000000000000E+00
-1.3655885572211683E-03 0.0000000000000000E+00 0.0000000000000000E+00
4.0967657318201995E-03 0.0000000000000000E+00 0.0000000000000000E+00
5.4623540682666614E-03 0.0000000000000000E+00 0.0000000000000000E+00
0.0000000000000000E+00 0.0000000000000000E+00 0.0000000000000000E+00
0.0000000000000000E+00 0.0000000000000000E+00 0.0000000000000000E+00
5.1209574149515494E-04 -1.3655886773915795E-03 0.0000000000000000E+00
1.7069857686018882E-04 0.0000000000000000E+00 -1.0241914729761090E-03
-1.0241915525371181E-03 0.0000000000000000E+00 3.4139717920607377E-04
-6.8279434812164720E-04 0.0000000000000000E+00 1.3655885338970169E-03
0.0000000000000000E+00 0.0000000000000000E+00 0.0000000000000000E+00
0.0000000000000000E+00 0.0000000000000000E+00 0.0000000000000000E+00
0.0000000000000000E+00 0.0000000000000000E+00 0.0000000000000000E+00
0.0000000000000000E+00 -1.3655886773915795E-03 0.0000000000000000E+00
1.7069857686018882E-04 0.0000000000000000E+00 0.0000000000000000E+00
-1.0241915525371181E-03 0.0000000000000000E+00 0.0000000000000000E+00
-6.8279434812164720E-04 0.0000000000000000E+00 1.3655885338970169E-03
C* COMPATIBILITY
0.0000000000000000E+00 0.0000000000000000E+00 0.0000000000000000E+00
0.0000000000000000E+00
-0.6666666666923697 0.0000000000000000E+00 0.0000000000000000E+00
0.0000000000000000E+00
0.0000000000000000E+00 0.0000000000000000E+00 0.0000000000000000E+00
0.0000000000000000E+00
-0.3333333333076302 0.0000000000000000E+00 0.0000000000000000E+00
0.0000000000000000E+00
0.0000000000000000E+00 0.1666666666538151 0.0000000000000000E+00
0.0000000000000000E+00
0.0000000000000000E+00 0.0000000000000000E+00 0.0000000000000000E+00
0.0000000000000000E+00
0.0000000000000000E+00 0.0000000000000000E+00 0.0000000000000000E+00
0.0000000000000000E+00
0.0000000000000000E+00 0.6666666666923698 0.0000000000000000E+00
0.0000000000000000E+00
0.0000000000000000E+00 0.0000000000000000E+00 0.0000000000000000E+00
0.0000000000000000E+00
0.0000000000000000E+00 0.0000000000000000E+00 0.0000000000000000E+00
0.0000000000000000E+00
0.0000000000000000E+00 0.0000000000000000E+00 0.0000000000000000E+00
0.0000000000000000E+00

0.0000000000000000E+00 0.0000000000000000E+00 0.0000000000000000E+00
0.0000000000000000E+00
0.0000000000000000E+00 0.1666666666538151 0.1666666910979775
0.0000000000000000E+00
0.0000000000000000E+00 0.0000000000000000E+00 0.0000000000000000E+00
0.0000000000000000E+00
0.0000000000000000E+00 0.0000000000000000E+00 0.0000000000000000E+00
0.0000000000000000E+00
0.0000000000000000E+00 0.0000000000000000E+00 0.6666666666923700
0.0000000000000000E+00
0.0000000000000000E+00 0.0000000000000000E+00 0.0000000000000000E+00
-1.0503229805754399E-17
0.0000000000000000E+00 0.0000000000000000E+00 0.0000000000000000E+00
0.6666666666923698
0.0000000000000000E+00 0.0000000000000000E+00 0.0000000000000000E+00
0.0000000000000000E+00
0.0000000000000000E+00 0.0000000000000000E+00 0.0000000000000000E+00
0.3333333333076302
0.0000000000000000E+00 0.0000000000000000E+00 0.1666666422096527
0.0000000000000000E+00
C* CPDDC
0.0000E+00 0.0000E+00 0.0000E+00 1.01 0.00 0.00 2.01
0.0000E+00 0.6350E-02 0.0000E+00 3.01 4.01 0.00 2.01
0.0000E+00 0.1270E-01 0.0000E+00 5.01 6.01 0.00 2.00
0.2540E-01 0.0000E+00 0.0000E+00 7.00 0.00 0.00 8.00
0.2540E-01 0.1270E-01 0.0000E+00 9.01 10.01 0.00 11.00
0.5080E-01 0.0000E+00 0.0000E+00 12.00 0.00 0.00 13.00
0.5080E-01 0.6350E-02 0.0000E+00 14.00 15.00 0.00 16.00
0.5080E-01 0.1270E-01 0.0000E+00 17.01 18.01 0.00 19.00
0.7620E-01 0.0000E+00 0.0000E+00 20.00 0.00 0.00 21.00
0.7620E-01 0.1270E-01 0.0000E+00 22.01 23.01 0.00 24.00
0.1016 0.0000E+00 0.0000E+00 25.01 0.00 0.00 0.00
0.1016 0.6350E-02 0.0000E+00 26.01 27.01 0.00 0.00
0.1016 0.1270E-01 0.0000E+00 28.01 29.01 0.00 0.00

```
*****************************************************************************
File: test1.eval
Contents: results (eigenfrequencies) of in vacuo modal analysis, short- and open-circuit
*****************************************************************************
 eigenfrequencies:
    mode      short circuit              open circuit       coupling constant
       1      ****rigid body mode****
       2  15086.20631077645      20069.01797555467      0.6594867
       3  44079.92520612262      44079.92520612253      0.0000000E+00
       4  65164.89844044350      68288.07488718008      0.2989630


*****************************************************************************
 File: test1.evec
Contents: results (eigenvectors) of in vacuo modal analysis
*****************************************************************************
 eigenvectors:
 mode:         2
  6.308654812022674      6.306605129685042     0.1832636726429298
  6.272137684637472     0.3653074505656848      3.819659078146684
  3.764273486300239     0.5484703063764521     1.27478038271535l0E-08
 1.2428974266584434E-08 0.3658927588020617      1.0782753170786425E-08
 0.7214488685849426     -3.819659317773155     -3.764273722653074
 0.5484703043306483     -6.308654813276728     -6.306605127434839
 0.1832636605255124     -6.272137671547285     0.3653074258032975
 mode:         3
  7.402756641663203      7.207390108823503     0.5137623123149291
  6.888321968944865     0.8510543482120636     -0.2219620512232982
 -4.4604025075992010E-02  1.239613664045570      -6.892353483214574
  -6.743102661502172     -5.8537570090980980E-08 -6.368083444316029
 -1.0591187375382722E-07 -0.2219612295016891     -4.4603315463997706E-02
  -1.239613608298329      7.402756006732976      7.207389489570373
 -0.5137621858480528      6.888321326079985     -0.8510541172456032

 mode:         4
  8.751935834806728      8.543794129661952      2.563842301653785
  8.704803043509511      4.194573778109600     -5.023077480264138
  -3.285145093938224     1.008760040772720      6.1928920338186464E-07
 6.0976793410806987E-07  -1.345691490344731      5.8775852703362183E-07
 -2.115969315200101      5.023077073975895      3.285144620300502
  1.008760360428842     -8.751936607357539     -8.543794835723848
  2.563842462285395     -8.704803701651505      4.194573997544881
```

****

File: test1.adisp

Contents: Results (displacements, admittance) of in vacuo harmonic analysis for frequencies: 15080 Hz to 15090 Hz

****

frequency	displacement at node 13 (DOF 20)	input admittance
15080.0	1.5847E-07	j* 4.4745E-04
15081.0	1.8889E-07	j* 5.3330E-04
15082.0	2.3376E-07	j* 6.5993E-04
15083.0	3.0673E-07	j* 8.6586E-04
15084.0	4.4570E-07	j* 1.2580E-03
15085.0	8.1496E-07	j* 2.3000E-03
15086.0	4.7534E-06	j* 1.3414E-02
15087.0	-1.2381E-06	j* -3.4937E-03
15088.0	-5.4808E-07	j* -1.5464E-03
15089.0	-3.5192E-07	j* -9.9288E-04
15090.0	-2.5916E-07	j* -7.3110E-04

displacements of all (21) mechanical DOFs at 15086 Hz:

-4.7811430663913224E-06 -4.7795896946743518E-06 -1.3889363115270528E-07
-4.7534686478735338E-06 -2.7686269107740883E-07 -2.8947905905332608E-06
-2.8528168817613219E-06 -4.1567192328096623E-07 -9.6607967926868083E-15
-9.4191773072928430E-15 -2.7729830545422933E-07 -8.1715961624483067E-15
-5.4676302267590435E-07 2.8947907721389695E-06 2.8528170608861054E-06
-4.1567192173054341E-07 4.7811430673416197E-06 4.7795896929689653E-06
-1.3889362196955851E-07 4.7534686379531221E-06 -2.7686267231131458E-07

```
********************************************************************************
File: ioelms.test1
Used by programs: imtest1.for, tvtest1.for, sc1test1.for, sc2test1.for
Used for analyses: in-water modal, radiation, & scattering
Contents:  connectivities of 2-D BE input geometry
Format: BE number, connectivity (end node, mid-side node, end node)
********************************************************************************
element  connectivity
   1    3    2    1          ! CHIEF connectivity for first boundary element; nodes match FE
nodes
   2    8    5    3
   3   13   10    8
   4   11   12   13


********************************************************************************
File: iomagc.test1
Used by programs: imtest1.for, tvtest1.for, sc1test1.for, sc2test1.for
Used for analyses: in-water modal, radiation, & scattering
Contents:  coordinates of 2-D BE geometry
Format: node number, CHIEF coordinates (r,z for axisymmetric input) in meters
********************************************************************************
   node       coordinates (r,z)
      1 0.0000000E+00 0.0000000E+00
      2 6.3499999E-03 0.0000000E+00
      3 1.2700000E-02 0.0000000E+00
      4 0.0000000E+00 2.5400000E-02
      5 1.2700000E-02 2.5400000E-02
      6 0.0000000E+00 5.0799999E-02
      7 6.3499999E-03 5.0799999E-02
      8 1.2700000E-02 5.0799999E-02
      9 0.0000000E+00 7.6200001E-02
     10 1.2700000E-02 7.6200001E-02
     11 0.0000000E+00 0.1016000
     12 6.3499999E-03 0.1016000
     13 1.2700000E-02 0.1016000
```

```
**************************************************************************
File:  test1.zrad
Created by CHIEF driver:  imtest1.for
Used by program: solver.f
Used for analysis: in-water modal
Format: free format, rows of [4x4] matrix output in continuous stream;
          real part, followed by imaginary part for each frequency
**************************************************************************

z at 15086 Hz:
(z11, z12, z13, z14, z21, z22, ...)
  33.36235     12.38048    -28.37361     6.247772     29.88015   !real
  688.4673     14.87788    -17.42863    -17.42968     14.87800
  688.4679     29.87747     6.248131    -28.37481     12.38085
  33.36449
  87.11289    -45.30119     39.41575    -2.040692    -42.34293   !imaginary
  628.3474    -222.3705     23.70166     23.70092    -222.3715
  628.3480    -42.34460    -2.040568     39.41782    -45.30351
  87.11729


**************************************************************************
file: test1.eigen
Contents: Results of in-water modal analysis (short-circuit and open-circuit)
**************************************************************************
***short-circuit condition***
 in vacuo eigenfrequency:       15086.00    Hz      !specified by user
 estimated in-fluid frequency:  14542.91    Hz
 estimated in-fluid frequency:  14527.44    Hz
 estimated in-fluid frequency:  14527.00    Hz
 final in-fluid frequency=      14527.00    Hz
 eigenvector for mode        2:
  6.122667498616763     6.120778596930935     0.1867223854669151
  6.089969720473470    0.3721452883633061     3.652879660524142
  3.603846916925769    0.5350962424432164     3.2925548803615677E-04
 3.9007126006716504E-04 0.3486120873087803     6.1165112576566541E-04
  0.6880883184194100    -3.650902233963054    -3.601799421571201
  0.5337224216988838    -6.119536692061142    -6.117756240853020
  0.1864765884848210    -6.087325961421187     0.3717102834825520

***open-circuit condition***
 in vacuo eigenfrequency:       20069.00    Hz      !specified by user
 estimated in-fluid frequency:  19554.84    Hz
```

estimated in-fluid frequency: 19555.78 Hz
final in-fluid frequency= 19555.78 Hz

eigenvector for mode 2:

5.664973461428674 5.663059663138925 4.3275910059854400E-02
5.605760143453185 8.8995358584600281E-02 3.967550797676853
3.870397436093299 0.4033650509505893 2.1362430566651417E-03
2.1315012677434694E-03 0.3588795190605135 2.0964455162545905E-03
0.7001311203995652 -3.964396590181886 -3.867097469826611
0.4033014519935140 -5.663606770147576 -5.661605778639167
4.2704098409301172E-02 -5.604014999920453 8.7842282633587176E-02

```
**************************************************************************
 File:  test1.zrad
Created by CHIEF driver:  imtest1.for
Used by program: solver.f
Used for analysis: radiation
Contents: mutual coupling matrix
Format: free format, rows of [4x4] matrix output in continuous stream;
          real part, followed by imaginary part
**************************************************************************
 z at 14513 Hz:
(z11, z12, z13, z14, z21, z22, ...)
  30.11400     16.85989    -33.70602     6.132087     32.97564   !real
  655.1061     41.39457    -21.15956    -21.16038     41.39480
  655.1065     32.97316     6.132379    -33.70755     16.86049
  30.11594
  84.19703    -42.01745     32.80411    -0.2495326    -38.25501   !imaginary
  630.0978    -218.1802     18.57932     18.57845     -218.1811
  630.0989    -38.25675    -0.2493281    32.80592     -42.01971
  84.20134


**************************************************************************
File: test1.fdisp
Contents: Results of radiation analysis (solver.f)
**************************************************************************
frequency  displ. at node 13 (DOF 20)    input admittance     velocity of BE 4 !magnitudes
  14505.00     3.8547827E-09              1.0507602E-05        3.5249183E-04
  14506.00     3.8552375E-09              1.0507952E-05        3.5255795E-04
  14507.00     3.8556340E-09              1.0508141E-05        3.5261869E-04
  14508.00     3.8559711E-09              1.0508167E-05        3.5267399E-04
  14509.00     3.8562509E-09              1.0508036E-05        3.5272405E-04
  14510.00     3.8564725E-09              1.0507747E-05        3.5276884E-04
  14511.00     3.8566350E-09              1.0507297E-05        3.5280822E-04
  14512.00     3.8567398E-09              1.0506689E-05        3.5284230E-04
  14513.00     3.8567864E-09              1.0505922E-05        3.5287102E-04
  14514.00     3.8567740E-09              1.0504992E-05        3.5289442E-04
  14515.00     3.8567030E-09              1.0503904E-05        3.5291241E-04

 displacements for frequency:    14513.00 Hz
(-2.1304333961902629E-10,3.8715065355311219E-09)
(-2.1297214412927733E-10,3.8703268737781678E-09)
(-6.9575930013952745E-12,1.1822650698951106E-10)
(-2.1190154488027883E-10,3.8509638190650367E-09)
```

(-1.3862775374303129E-11,2.3564279234432265E-10)
(-1.2705214013221493E-10,2.3089515692841767E-09)
(-1.2534862742532883E-10,2.2779839702285985E-09)
(-1.9512554193810485E-11,3.3830163692865427E-10)
(1.8526031105089533E-15,4.1118665726134529E-15)
(1.8407099609733817E-15,4.0947208137145191E-15)
(-1.2585175607466857E-11,2.2050545391635330E-10)
(1.8017087245688336E-15,4.0405813150657165E-15)
(-2.4846170909699200E-11,4.3524462477540238E-10)
(1.2705512148497511E-10,-2.3089449154498963E-09)
(1.2535155850429226E-10,-2.2779774068023876E-09)
(-1.9512243155823768E-11,3.3830210300575184E-10)
(2.1304456463404699E-10,-3.8715033528952930E-09)
(2.1297337137315276E-10,-3.8703236965782366E-09)
(-6.9573998349144432E-12,1.1822685724585442E-10)
(2.1190278191208932E-10,-3.8509606577081800E-09)
(-1.3862397846311565E-11,2.3564348125726239E-10)
input admittance= (1.0459494569941213E-05,9.8659280032537155E-07)
normal surface velocities:
(3.5233829438449920E-04,1.9387933706123919E-05)
(-3.0762135273395744E-05,-1.7745031756620410E-06)
(-3.0762174522110157E-05,-1.7744785538891844E-06)
(3.5233800514397932E-04,1.9388045913389083E-05)

****

File: test1.ffpb
Created by CHIEF driver: tvtest1.for
Contents: radiated far-field pressure vs. radian angle for 14513 Hz

****

test1.ffpb at 14513.00 Hz:

angle	pressure
0.0000000	(3.8649984E-02,2.719644)
0.1745329	(0.2063564,2.807488)
0.3490658	(0.7301321,2.958420)
0.5235988	(1.567469,2.856166)
0.6981317	(2.385230,2.151870)
0.8726646	(2.537316,0.8955843)
1.047198	(1.632260,-0.1474671)
1.221730	(0.2933735,-8.5124344E-02)
1.396263	(-0.2281480,0.8512533)
1.570796	(0.3086637,1.301302)
1.745329	(0.6387287,0.6072051)

1.919862	(-0.2241971,-0.2075152)
2.094395	(-1.637815,6.0186878E-02)
2.268928	(-2.375069,1.264551)
2.443461	(-2.105991,2.425816)
2.617994	(-1.364877,2.958306)
2.792527	(-0.7160859,2.961812)
2.967060	(-0.3569838,2.792289)
3.141593	(-0.2511753,2.708247)
3.316126	(-0.3569838,2.792289)
3.490659	(-0.7160859,2.961812)
3.665191	(-1.364877,2.958306)
3.839724	(-2.105991,2.425816)
4.014257	(-2.375069,1.264550)
4.188790	(-1.637813,6.0186192E-02)
4.363323	(-0.2241967,-0.2075149)
4.537856	(0.6387290,0.6072056)
4.712389	(0.3086635,1.301302)
4.886922	(-0.2281480,0.8512537)
5.061455	(0.2933728,-8.5124075E-02)
5.235988	(1.632260,-0.1474675)
5.410521	(2.537316,0.8955829)
5.585053	(2.385231,2.151869)
5.759586	(1.567470,2.856165)
5.934119	(0.7301332,2.958420)
6.108652	(0.2063571,2.807488)
6.283185	(3.8649984E-02,2.719644)

****

File: test1.patt
Created by CHIEF driver: tvtest1.for
Contents: normalized pattern (far-field) vs. radian angle for 14513 Hz

****

test1.patt at 14513 Hz:

frequency		theta	phi	20log|p/pnorm)
14513.00	0.000000	0.0000000E+00	0.0000000E+00	
14513.00	10.00000	0.0000000E+00	0.2986406	
14513.00	20.00000	0.0000000E+00	0.9868609	
14513.00	30.00000	0.0000000E+00	1.567935	
14513.00	40.00000	0.0000000E+00	1.445623	
14513.00	50.00000	0.0000000E+00	-9.3706086E-02	
14513.00	60.00000	0.0000000E+00	-4.400024	
14513.00	70.00000	0.0000000E+00	-18.99164	

14513.00	80.00000	0.0000000E+00	-9.788676
14513.00	90.00000	0.0000000E+00	-6.165837
14513.00	100.0000	0.0000000E+00	-9.788740
14513.00	110.0000	0.0000000E+00	-18.99105
14513.00	120.0000	0.0000000E+00	-4.399961
14513.00	130.0000	0.0000000E+00	-9.3706608E-02
14513.00	140.0000	0.0000000E+00	1.445587
14513.00	150.0000	0.0000000E+00	1.567864
14513.00	160.0000	0.0000000E+00	0.9867527
14513.00	170.0000	0.0000000E+00	0.2985016
14513.00	180.0000	0.0000000E+00	-1.5479948E-04
14513.00	190.0000	0.0000000E+00	0.2985016
14513.00	200.0000	0.0000000E+00	0.9867527
14513.00	210.0000	0.0000000E+00	1.567863
14513.00	220.0000	0.0000000E+00	1.445587
14513.00	230.0000	0.0000000E+00	-9.3707129E-02
14513.00	240.0000	0.0000000E+00	-4.399970
14513.00	250.0000	0.0000000E+00	-18.99106
14513.00	260.0000	0.0000000E+00	-9.788733
14513.00	270.0000	0.0000000E+00	-6.165837
14513.00	280.0000	0.0000000E+00	-9.788672
14513.00	290.0000	0.0000000E+00	-18.99166
14513.00	300.0000	0.0000000E+00	-4.400025
14513.00	310.0000	0.0000000E+00	-9.3708180E-02
14513.00	320.0000	0.0000000E+00	1.445622
14513.00	330.0000	0.0000000E+00	1.567935
14513.00	340.0000	0.0000000E+00	0.9868609
14513.00	350.0000	0.0000000E+00	0.2986417
14513.00	360.0000	0.0000000E+00	0.0000000E+00

```
**********************************************************************
File:  test1.zrad
Created by CHIEF driver: sc1test1.for
Used by program: solver.f
Used for analysis: scattering
Contents: mutual coupling matrix
Format: free format, rows of [4x4] matrix output in continuous stream;
        real part, followed by imaginary part
**********************************************************************
z at 19374 Hz:
(z11, z12, z13, z14, z21, z22, ...)
  62.08010    -28.36123     29.64417    -6.521568    -1.002919   !real
  886.1635    -138.9735     30.29880     30.29756    -138.9745
  886.1646    -1.006932    -6.521462     29.64565    -28.36270
  62.08373
  100.2645    -41.82556     38.70781    -7.121194    -53.42942   !imaginary
  537.5993    -126.0520     21.04491     21.04629    -126.0526
  537.5999    -53.42981    -7.121778     38.70961    -41.82757
  100.2690


**********************************************************************
File:  test1.pinc
Created by CHIEF driver: sc1test1.for
Used by program: solver.f
Used for analysis: scattering
**********************************************************************
test1.pinc at 19374 Hz:
 1.6585976E-04 -1.4249036E-04   !real,imag
-3.9952548E-04  5.9578021E-04
 4.2084706E-04 -5.5839802E-05
-1.1910323E-04  2.1365639E-04


**********************************************************************
File: test1.ffpa
Created by CHIEF driver: sc1test1.for
Contents: rigid scattered pressures
**********************************************************************
test1.ffpa at 19374 Hz:
    theta                          p_r
 0.0000000E+00 (1.9917404E-03,-1.2901641E-02)
 0.1745329    (1.1250330E-03,-1.3349861E-02)
```

0.3490658	(-1.6777457E-03,-1.3714730E-02)
0.5235988	(-5.7329829E-03,-1.1320981E-02)
0.6981317	(-7.0530595E-03,-4.3074954E-03)
0.8726646	(-3.8230792E-04,2.9844809E-03)
1.047198	(1.1124386E-02,9.5268525E-04)
1.221730	(1.4162721E-02,-1.0273563E-02)
1.396263	(5.4635201E-03,-1.5681677E-02)
1.570796	(-3.8920590E-04,-9.8560723E-03)
1.745329	(2.0337133E-03,-5.4256851E-03)
1.919862	(2.4907449E-03,-5.9257564E-03)
2.094395	(5.8426522E-06,-2.9836274E-03)
2.268928	(1.9156323E-03,2.5573976E-03)
2.443461	(7.1422746E-03,3.9579412E-03)
2.617994	(9.9874260E-03,1.2264601E-03)
2.792527	(9.3676448E-03,-1.6260156E-03)
2.967060	(7.6950951E-03,-2.8826012E-03)
3.141593	(6.9436021E-03,-3.1272531E-03)
3.316126	(7.6950956E-03,-2.8826017E-03)
3.490659	(9.3676466E-03,-1.6260152E-03)
3.665191	(9.9874269E-03,1.2264589E-03)
3.839724	(7.1422774E-03,3.9579393E-03)
4.014257	(1.9156290E-03,2.5573932E-03)
4.188790	(5.8435835E-06,-2.9836320E-03)
4.363323	(2.4907475E-03,-5.9257555E-03)
4.537856	(2.0337135E-03,-5.4256851E-03)
4.712389	(-3.8920538E-04,-9.8560732E-03)
4.886922	(5.4635154E-03,-1.5681677E-02)
5.061455	(1.4162722E-02,-1.0273567E-02)
5.235988	(1.1124387E-02,9.5268525E-04)
5.410521	(-3.8230186E-04,2.9844849E-03)
5.585053	(-7.0530563E-03,-4.3074829E-03)
5.759586	(-5.7329843E-03,-1.1320977E-02)
5.934119	(-1.6777508E-03,-1.3714727E-02)
6.108652	(1.1250293E-03,-1.3349862E-02)
6.283185	(1.9917404E-03,-1.2901641E-02)

****

File: test1.tgtr

Contents: target strength of rigid surface

****

test1.tgtr at 19374 Hz:

theta	20log|p/ainc|
0.0000000E+00	-37.68481
10.00000	-37.45973
20.00000	-37.19174
30.00000	-37.93089
40.00000	-41.65578
50.00000	-50.43194
60.00000	-39.04274
70.00000	-35.14096
80.00000	-35.59461
90.00000	-40.11916
100.0000	-44.73996
110.0000	-43.83856
120.0000	-50.50509
130.0000	-49.90977
140.0000	-41.76022
150.0000	-39.94593
160.0000	-40.43848
170.0000	-41.70543
180.0000	-42.36622
190.0000	-41.70543
200.0000	-40.43847
210.0000	-39.94593
220.0000	-41.76021
230.0000	-49.90979
240.0000	-50.50507
250.0000	-43.83856
260.0000	-44.73996
270.0000	-40.11916
280.0000	-35.59461
290.0000	-35.14096
300.0000	-39.04274
310.0000	-50.43193
320.0000	-41.65579
330.0000	-37.93089
340.0000	-37.19175
350.0000	-37.45973
360.0000	-37.68481

****

File: test1.fpdisp

Contents: results of solver for incident plane wave of amplitude 1

****

frequency	displ. at node 13 (DOF 20)	sensor voltage	velocity of BE 4 !magnitudes
19350.00	3.2981807E-12	2.1971637E-02	4.0298667E-07
19351.00	3.2986049E-12	2.1975905E-02	4.0305946E-07
19352.00	3.2989991E-12	2.1979976E-02	4.0312855E-07
19353.00	3.2993959E-12	2.1984069E-02	4.0319804E-07
19354.00	3.2997654E-12	2.1987984E-02	4.0326418E-07
19355.00	3.3001288E-12	2.1991862E-02	4.0332958E-07
19356.00	3.3004649E-12	2.1995561E-02	4.0339162E-07
19357.00	3.3007815E-12	2.1999136E-02	4.0345131E-07
19358.00	3.3010981E-12	2.2002712E-02	4.0351100E-07
19359.00	3.3013880E-12	2.2006113E-02	4.0356741E-07
19360.00	3.3016649E-12	2.2009434E-02	4.0362227E-07
19361.00	3.3019136E-12	2.2012569E-02	4.0367368E-07
19362.00	3.3021587E-12	2.2015681E-02	4.0372464E-07
19363.00	3.3023890E-12	2.2018699E-02	4.0377375E-07
19364.00	3.3025880E-12	2.2021515E-02	4.0381912E-07
19365.00	3.3027778E-12	2.2024274E-02	4.0386337E-07
19366.00	3.3029369E-12	2.2026829E-02	4.0390381E-07
19367.00	3.3030826E-12	2.2029299E-02	4.0394264E-07
19368.00	3.3032299E-12	2.2031782E-02	4.0398166E-07
19369.00	3.3033435E-12	2.2034047E-02	4.0401659E-07
19370.00	3.3034298E-12	2.2036131E-02	4.0404817E-07
19371.00	3.3035035E-12	2.2038136E-02	4.0407821E-07
19372.00	3.3035629E-12	2.2040047E-02	4.0410646E-07
19373.00	3.3036089E-12	2.2041872E-02	4.0413312E-07
19374.00	3.3036206E-12	2.2043476E-02	4.0415560E-07
19375.00	3.3036145E-12	2.2044960E-02	4.0417586E-07
19376.00	3.3035957E-12	2.2046363E-02	4.0419459E-07
19377.00	3.3035556E-12	2.2047630E-02	4.0421071E-07
19378.00	3.3034996E-12	2.2048792E-02	4.0422492E-07
19379.00	3.3034187E-12	2.2049794E-02	4.0423606E-07
19380.00	3.3033123E-12	2.2050625E-02	4.0424405E-07

displacements for frequency: 19374.00

(3.2341147210942354E-12,-1.2999930066497998E-12)

(3.2305064026933850E-12,-1.2974408864684124E-12)

(7.5976179041457552E-14,-7.9165705439804396E-14)

(3.1916246005606838E-12,-1.2798377481679852E-12)

(1.4942339805429891E-13,-1.5473145983208721E-13)
(1.8433210525009184E-12,-4.2382228044645483E-13)
(1.8038247701964939E-12,-4.1738899917617075E-13)
(3.1634399279006147E-13,-2.0862316386218662E-13)
(-5.5949984186890707E-13,6.7170263847811024E-13)
(-5.5196220968681404E-13,6.6438415138578987E-13)
(1.9781928230813311E-13,-9.2882235817899985E-14)
(-5.2750955598344921E-13,6.4099380578619700E-13)
(3.8607729251716107E-13,-1.8158333402442045E-13)
(-2.5599311194030207E-12,1.3204718591566993E-12)
(-2.4935326766295697E-12,1.2850137213561425E-12)
(1.3185896504139117E-13,-2.6291315823638508E-14)
(-3.0986974040516779E-12,1.2096105914788331E-12)
(-3.1003292866172060E-12,1.2106486238968786E-12)
(-2.4429409346455475E-14,2.9665456079069591E-14)
(-3.0768225500434691E-12,1.2029429989090884E-12)
(-4.3757521114142953E-14,5.5608619591631080E-14)
normal surface velocities:
(-1.5722376749801472E-07,-3.9167325815173783E-07)
(2.3753817172157888E-08,3.6536917319508724E-08)
(4.6894624744215191E-09,1.7645956870381629E-08)
(-1.4706011694543922E-07,-3.7645061201160950E-07)
phi= (2.0377204074682805E-02,-8.4073991802671166E-03)

****

File: test1.ffpb

****

test1.ffpb at 19374 Hz:

theta	p_e
0.0000000	(-1.5188868E-03,2.7887448E-04)
0.1745329	(-1.1149275E-03,4.6940567E-04)
0.3490658	(1.3935077E-04,7.2098069E-04)
0.5235988	(1.9880498E-03,1.7045595E-04)
0.6981317	(3.1698248E-03,-1.8154135E-03)
0.8726646	(2.0162207E-03,-4.0380303E-03)
1.047198	(-7.2011171E-04,-3.6955094E-03)
1.221730	(-1.2761313E-03,-7.2704151E-04)
1.396263	(1.2349433E-03,5.6220114E-04)
1.570796	(2.3281910E-03,-1.1862505E-03)
1.745329	(5.7865679E-04,-1.4961411E-03)
1.919862	(3.2634105E-04,9.2571252E-04)
2.094395	(2.7811909E-03,2.0481837E-03)

2.268928	(4.4772262E-03,5.5927777E-04)
2.443461	(3.9088042E-03,-9.4699109E-04)
2.617994	(2.6334615E-03,-9.3149487E-04)
2.792527	(1.9682981E-03,-5.2091666E-05)
2.967060	(1.8918687E-03,7.1960792E-04)
3.141593	(1.9364059E-03,9.9493971E-04)
3.316126	(1.8918687E-03,7.1960804E-04)
3.490659	(1.9682981E-03,-5.2091666E-05)
3.665191	(2.6334617E-03,-9.3149522E-04)
3.839724	(3.9088032E-03,-9.4699167E-04)
4.014257	(4.4772262E-03,5.5927958E-04)
4.188790	(2.7811886E-03,2.0481842E-03)
4.363323	(3.2634076E-04,9.2571147E-04)
4.537856	(5.7865749E-04,-1.4961415E-03)
4.712389	(2.3281910E-03,-1.1862498E-03)
4.886922	(1.2349444E-03,5.6220090E-04)
5.061455	(-1.2761303E-03,-7.2704017E-04)
5.235988	(-7.2011235E-04,-3.6955089E-03)
5.410521	(2.0162184E-03,-4.0380321E-03)
5.585053	(3.1698258E-03,-1.8154167E-03)
5.759586	(1.9880508E-03,1.7045537E-04)
5.934119	(1.3935263E-04,7.2098034E-04)
6.108652	(-1.1149257E-03,4.6940672E-04)
6.283185	(-1.5188868E-03,2.7887448E-04)

****

File: test1.tgtt

Contents: rigid+elastic target strength

****

theta	20log((p$_r$+p$_c$)/ainc)
0.0000000	-37.97082
0.1745329	-37.80137
0.3490658	-37.66486
0.5235988	-38.58994
0.6981317	-42.79266
0.8726646	-54.22550
1.047198	-39.36396
1.221730	-35.42001
1.396263	-35.63094
1.570796	-39.00690
1.745329	-42.61726
1.919862	-44.82323

2.094395	-50.63355
2.268928	-42.96001
2.443461	-38.82092
2.617994	-37.97583
2.792527	-38.81670
2.967060	-40.15075
3.141593	-40.78827
3.316126	-40.15075
3.490659	-38.81670
3.665191	-37.97583
3.839724	-38.82092
4.014257	-42.96002
4.188790	-50.63355
4.363323	-44.82323
4.537856	-42.61726
4.712389	-39.00690
4.886922	-35.63094
5.061455	-35.42001
5.235988	-39.36396
5.410521	-54.22549
5.585053	-42.79267
5.759586	-38.58994
5.934119	-37.66486
6.108652	-37.80137
6.283185	-37.97082

APPENDIX D

FORTRAN PROGRAMS

This appendix contains the source code for the external solver program, solver.f, and for the CHIEF driver programs in the following order: imtest1.for, tvtest1.for, sc1test1.for, and sc2test1.for. The program imtest1.for is used to compute the [Z] matrix for the in-water modal analysis, and for the radiation analysis, while sc1test1.for computes [Z] and $\{p_{sr}\}$ for the scattering analysis. The programs tvtest1.for and sc1test1.for compute the radiated and elastic scattered pressures, respectively.

```
**************************************************************************
File: solver.f
Contents: Fortran program for reading in matrices and solving linear systems
**************************************************************************

c*****************************************************************
c  solver.f
c          SUMMER 95
c*****************************************************************
c This program determines the modal or harmonic solution of the coupled
c  Boundary Element/Finite Element problem.  For any of these analyses,
c  the user must provide the mass, stiffness, and compatibility matrices
c  calculated by a finite element program.  If an in-fluid analysis is
c  desired, the user must compute the mutual coupling matrix in CHIEF.
c  We assume the stiffness matrix is real, but the program can be
c  modified to include material losses.  The types of analysis
c  that are included:
c   Modal of in vacuo structure - elastic or short circuit piezoelectric
c    and open circuit piezoelectric.
c   Modal of in-fluid structure - elastic or short circuit piezoelectric.
c     This is an iterative solution.
c   Harmonic of in vacuo structure - piezoelectric with applied potential.
c   Harmonic of in-fluid structure - piezoelectric with applied potential, or
c                              piezoelectric with incident plane wave.
c  A good reference on these analyses is:  "Application of the Finite Element/
c  Boundary Element Approach to the Analysis of Radiation and Scattering from
c  Fluid-Loaded Elastic and Piezoelectric Structures," by M.D. McCollum and
c  R.E. Mongomery, NUWC Technical Document, Newport, RI.
c*****************************************************************************
c*****************************************************************************
c INPUT for in vacuo analysis
c          nelecc (PARAMETER) number of free external potentials (not ground)
c          nareac (PARAMETER) number of boundary elements
c          nmodes (PARAMETER) number of nodes
```

```
c          kuutmp (nmodes *(nmodes+1)/2)--elastic stiffness with internal
c            potentials statically condensed--Finite element output--upper
c       triangular matrix
c          kuevec(nmodes,nelecc)--piezoelectric stiffness
c          mastmp(nmodes*(nmodes+1)/2)--mass--Finite element output--upper
c       triangular matrix
c          kee--dielectric stiffness-external potentials--Finite element
c       output. Note: this program assumes one free external potential, so
c       kee is a scalar. For more than one, kee must be dimensioned
c       appropriately.

c INPUT for in-water analysis
c          zr(nareac,nareac)real part of the mutual coupling matrix--CHIEF
c          zi(nareac,nareac)imag part of the mutual coupling matrix--CHIEF

c INPUT for reordering displacements
c          nelin(PARAMETER)number of internal potentials
c          ncpddc (PARAMETER) number of columns in D.O.F. table
c          cpddc(nnodes,ncpddc) D.O.F. correspondence table

c OUTPUT
c          w(nmodes), w1(nmodes) short-circuit, open-circuit eigenvalues
c          p in-fluid eigenvalue (short circuit)
c          x(nmodes,nelecs) in vacuo displacements
c      cx(nmodes,nelecs) complex in-fluid displacements
c*************************************************************************

c Elastic stiffness (with internal potentials statically condensed):
c       kuutmp,dpk,dpkk
c Mass:   mastmp,dpm,dpmm
c Piezoelectric stiffness (external potentials):
c       kuevec,ckuevec
c Dielectric stiffness (external potentials):
c       kee
c Compatibility:
c       bigx,cbigx,b,br
c Mutual coupling:
c       real: zr
c       imaginary: zi
c DOF correspondence:
c       cpddc
c Eigenvalues:
c       eval
```

```
c Eigenvectors:
c       evec
c Displacements:
c       x,cx
c Dynamic stiffness (K- 2M):
c       rkm,crkm


**********************************************************************************
**********************************************************************************
*                    BEGINNING OF PROGRAM                                        *
**********************************************************************************
**********************************************************************************
c The user must set the following parameters in order for the matrices to be dimensioned
c correctly.  The program will read the correct values from the .matrices file and check
c that they are consistent with the parameter statements.
c*********************************************************************************
c*********************************************************************************
c If there are no electrodes (elastic materials only), the user must set nelecc=0
c and nelecs=1, otherwise, set nelecs and nelecc to the number of free electrodes
c
c piezoelectric problem
      PARAMETER (nelecc = 1)        !number of free external potentials
      PARAMETER (nelecs = nelecc)
c elastic problem
c     parameter(nelecc = 0)
c     parameter(nelecs = 1)         !avoid zero dimension
c*********************************************************************************
c If there are no wetted surfaces, the user must set nareac=0
c and nareas=1, otherwise, set nareac and nareas to the number of surfaces
c
c in-fluid problem
      PARAMETER (nareac = 4)        !number of boundary elements
      PARAMETER (nareas = nareac)
c in vacuo problem
c     parameter(nareac = 0)
c     parameter(nareas = 1)         !avoid zero dimension
c*********************************************************************************

      PARAMETER (nmodes = 21)       !number of mechanical dofs
      PARAMETER (nelin  = 7)        !number of internal potentials
      PARAMETER (nnodes = 13)       !number of nodes
      PARAMETER (ncpddc = 7)        !number of columns in atila dof table
```

```
c these parameters are computed by the code
      parameter (nmode2=nmodes*(nmodes+1)/2) !number of terms in the upper diag.
      parameter (ndofs=nmodes+nelin+nelecc)  !total number of dofs (mech+elec)
c*************************************************************************

c  system is used to execute shell commands from within the fortran program.
      external system

c stiffness
      real*8 kuutmp(nmode2),dpk(nmodes, nmodes),dpkk(nmodes, nmodes)

c mass
      real*8 mastmp(nmode2),dpm(nmodes, nmodes),dpmm(nmodes, nmodes)

c dynamic stiffness
      real*8 rkm(nmodes,nmodes)
      complex*16 crkm(nmodes,nmodes)

c piezoelectric stiffness
      complex*16 ckuevec(nmodes,nelecs)
      real*8 kuevec(nmodes,nelecs)
      real*8 x(nmodes,nelecs)

c dielectric stiffness (give appropriate dimensions if more than 1 ext. electrode)
      real*8 kee
      complex*16 ckee

c mutual coupling
      complex*16 zrad(nareas,nareas)
      real*8 zr(nareas,nareas)
      real*8 zi(nareas,nareas)

c incident surface pressure
      real*8 pincr,pinci
      complex*16 pinc(nareas,nelecs)

c compatibility, area
      complex*16 cbigx(nmodes,nareas)
      real*8 bigx(nmodes,nareas)
      real*8 areas(nareas)

c dof correspondence
      real*4 cpddc(nnodes,ncpddc)
```

```
c eigenvalues, eigenvectors
      real*8 w(nmodes),w1(nmodes)

c displacements
      complex*16 cx(nmodes,nelecs)  !also x (above)

c external potential
      complex*16 phi(nelecs,nelecs)
      real*8 phir(nelecs,nelecs)
c excitation frequency
      real*8 omega,omega2

c temporary variables and arrays
      complex*16 b(nmodes,nareas)
      real*8 br(nmodes,nareas)
      real*8 ferr(nelecs),berr(nelecs)
      real*8 newfreq
      real*8 xtemp(6)
      real*8 work(nmodes*3)
      real*8 alpha,beta
      real*8 rwork(nmodes)
      real*8 af(nmodes,nmodes),r(nmodes),c(nmodes),rcond
      real*8 l(nmodes, nmodes)
      real*8 l1(nmodes, nmodes)
      real*8 l2(nmodes, nmodes)
      real*8 l3(nmodes, nmodes)
      real*4 zero
      complex*16 aff(nmodes,nmodes)
      complex*16 xalpha,xbeta,cwork(nmodes*3)
      integer err
      integer case_start,case_end,case_step
      integer doeigen,dohar
      integer iwork(nmodes),ipiv(nmodes)
      integer ipot(ndofs),inew(ndofs),iold(ndofs)
      character*10 stfreq                    ! used to pass frequency to chief run
      character*1 jobz,uplo,transa,transb,fact,trans
      character*1 equed
      character*1  answer
      character*3 fchief
      character*24 filein
      character*80 buffer
      character*100 string
      character*255 adum
```

```
      pi = acos(-1.0)
c-----------------------------------------------------------------------
c  get name of input files (without extension)
c-----------------------------------------------------------------------
      print *,'enter jobname without extension'
      read(5,'(q,a)')lfile,filein(1:lfile)
c-----------------------------------------------------------------------
c  open matrix file.
c-----------------------------------------------------------------------
      err=0
      open(unit=1,file=filein(1:lfile)//'.matrices',type='old',
     +   iostat=err)
      if(err.ne.0)then
        print *,'matrices file not found: ',filein//'.matrices'
        stop
      endif
c-----------------------------------------------------------------------
c Get  analysis information interactively.
c-----------------------------------------------------------------------
      ians=0
17     print *,'     '
      print *,'modal or harmonic analysis? (m/h)'
      read(5,'(q,a)')lsize,answer
      if((answer.ne.'h').and.(answer.ne.'H') .and.
     +   (answer.ne.'m').and.(answer.ne.'M'))goto 17
      if((answer.eq.'h').or.(answer.eq.'H'))then
        dohar=1
        print *,'enter freq (min,max,step) (hz)'
        read (5,*)fr1,fr2,delfr
      else
        doeigen=1
55       print *,'enter loop of eigenvectors to save'
        print *,'(case_start,case_end)'
        read (*,*)case_start,case_end
        case_step=1
        if(case_start.lt.1) case_start=1
        if((case_end.gt.nmodes).or.(case_end.lt.case_start))then
          print *,'ending case is out of range. Try again.'
          go to 55
        end if
      end if
      if (nareac .gt. 0)then
```

```
      print *,'include fluid loading',
     +        ' in the solution? (y/n) '
      read(5,'(q,a)')lsize,answer
      if((answer.eq.'y').or.(answer.eq.'Y'))then
       ians=1
       if(dohar.eq.1)then
        print *,'radiation or scattering analysis? (r/s)'
        read(5,'(q,a)')lsize,answer
        if((answer.eq.'r').or.(answer.eq.'R'))then
         irad=1
        else
         irad=0
        endif
       endif
      endif
     endif
c*******************************************************************************
c  The following code reads in the upper triangular stiffness and
c  mass matrix, and the compatibility matrix in the ATILA Finite Element
c  format.  If the output from another finite element program is being
c  used, the format of the following read statements will most likely
c  need to be altered
c*******************************************************************************
      read(1,11)adum
11     format(a)
      read(1,11)adum
c size of problem
      print *,'reading parameters from 1st line of file'
      read(1,*) imeca, ielin, ielex, nsurf, ip, ix

c check for consistency of values with parameter statements
      if( (imeca.ne.nmodes).or.(ielin.ne.nelin).or.
     +   (ielex.ne.nelecc).or.(nsurf.ne.nareac).or.
     +   (ip.ne.nnodes).or.(ix.ne.ncpddc) )then
        print *,'header values in .matrices file not
     + consistent with parameter statements'
        print *,'                file:   program:'
        print *,'mech dofs: ',imeca,nmodes
        print *,'int pot:   ',ielin,nelin
        print *,'ext pot:   ',ielex,nelecc
        print *,'be:        ',nsurf,nareac
        print *,'nodes:     ',ip,nnodes
        print *,'cols:      ',ix,ncpddc
```

```
        stop
      endif
c kuu
      print *,'reading kuu'
      read(1,11)adum
      read(1,*)kuutmp
c kue
      if(nelecc.gt.0)then
       print *,'reading kue'
       read(1,11)adum
       do i=1,nmodes
         read(1,*)(kuevec(i,j),j=1,nelecc)
       end do
c kee
       read(1,11)adum
       print *,'reading kee'
       read(1,*)kee
      end if
c mass
      read(1,11)adum
      print *,'reading mass matrix'
      read(1,*)mastmp
c compatibility
      if(nareac .gt. 0)then
        print *,'reading compatibility'
        read(1,11)adum
        do i=1,nmodes
          read(1,*,err=15)(bigx(i,j),j=1,nareac)
        end do
      endif
c dof correspondence
15     print *,'reading dof table'
      read(1,11)adum
      do i=1,nnodes
        read(1,*)(cpddc(i,j),j=1,ncpddc)
      end do
3     continue
      close(1)
c*******************************************************************************
c  END OF ATILA Finite Element Matrices Read
c*******************************************************************************

c-------------------------------------------------------------------
```

```
c  set up symmetric stiffness and mass matrices from tmp vectors
c-----------------------------------------------------------------------
      print *,'      '
      print *,'forming symmetric stiffness and mass matrices'
      num=0
      do i = 1, nmodes
       do j = 1, nmodes
        if(j.le.i)then
          num=num+1
          dpm(i, j) = mastmp(num)
          dpm(j, i) = dpm(i,j)
          dpmm(i, j) = mastmp(num)
          dpmm(j, i) = dpmm(i,j)
          dpk(i, j) = kuutmp(num)
          dpk(j, i) = dpk(i,j)
          dpkk(i, j) = kuutmp(num)
         dpkk(j, i) = dpkk(i,j)
        end if
       end do
      end do

c if fluid loading included set up complex matrices
      if((nareac .gt. 0).and.(dohar.eq.1).and.(ians.eq.1))then
        print *,'forming complex matrices'
        do i=1,nmodes
          do j=1,nelecc
            ckuevec(i,j)=dcmplx(kuevec(i,j),0.d+0)
          enddo
        enddo
        do i=1,nmodes
          do j=1,nareac
            cbigx(i,j)=dcmplx(bigx(i,j),0.d+0)
          end do
        end do
      endif
      do i=1,nmodes
        if(dpm(i,i).le.0.d0)then
          print *,'mass not positive at: ',i,dpm(i,i)
          go to 999
        end if
        if(dpk(i,i).le.0.d0)then
          print  *,'stiffness not positive at: ',i,dpk(i,i)
          go to 999
```

```
        end if
      end do

c*************************************************************************
c*************************************************************************
c Case I.  In vacuo eigensolution.
c*************************************************************************
c*************************************************************************
      if (doeigen .eq. 1 .and. ians .eq. 0)then
       print *,'computing elastic or short-circuit eigenvalues
     +and eigenvectors'

c open file for output
      open(unit=98,file=filein(1:lfile)//'.eval',status='unknown')
      open(unit=99,file=filein(1:lfile)//'.evec',status='unknown')

c short-circuit or elastic eigensolution
c put elastic or short-circuit eigenvalues in w and eigenvectors in dpk
c-------------------------------------------------------------------
c  LAPACK Routine that computes all eigenvalues and eigenvectors
c-------------------------------------------------------------------
      jobz='v'
      uplo='u'
      lwork=3*nmodes
      call dsygv(1,jobz,uplo,nmodes,dpk,nmodes,dpm,nmodes,
     +   w,work,lwork,info)
      if(nelecc .eq. 0)then
        print *,'frequencies'
        write(98,*)'eigenfrequencies: '
        write(99,*)'eigenvectors:'
        do i=case_start,case_end
          if (w(i) .gt. 0)then
            print*, i,dsqrt(w(i))/(2.*pi)
            write(98,*)i,dsqrt(w(i))/(2.*pi)
            write(99,*)'mode: ',i
            write(99,*)(dpk(ii,i),ii=1,nmodes)
          else
            write(99,*)i,'      ****rigid body mode****'
          endif
        end do
        close(98)
        close(99)
        print *,'Results are in file: ',filein(1:lfile)//
```

```
     +'.eval/.evec'
        else
         print *,'computing open-circuit eigenvalues and eigenvectors'

c modification to stiffness matrix for open circuit solution
c multiply (-1/kee) by kuevec by transpose(kuevec), add to dpkk,
c   and put result in dpkk
c-----------------------------------------------------------------------
c LAPACK routine that does a matrix multiplication
c-----------------------------------------------------------------------
         transa='n'
         transb='t'
         m=nmodes
         k=nelecc
         n=nmodes
         alpha=-1.0/kee
         beta=1.0
         lda=nmodes
         ldb=nmodes
         ldc=nmodes
         call dgemm(transa,transb,m,n,k,alpha,kuevec, lda,
     +     kuevec,ldb,beta,dpkk,ldc)

c open-circuit eigensolution
c put o-c eigenvalues in w1 and eigenvectors in dpkk
c note: only s-c eigenvectors are printed out
c-----------------------------------------------------------------------
c  LAPACK Routine that computes all eigenvalues and eigenvectors
c-----------------------------------------------------------------------
         call dsygv(1,jobz,uplo,nmodes,dpkk,nmodes,dpmm,nmodes,
     +     w1,work,lwork,info)

         print *,' '
         print *,'     mode     short circuit          open  circuit'
         print *,'     ----     -------------          -------------'
         write(98,*)'eigenvalues:'
         write(98,*)'     mode     short circuit          open
    +circuit    coupling constant'
         write(99,*)'eigenvectors:'
         do i=case_start,case_end
          if (w1(i) .gt. 0 .and. w(i) .gt. 0)then
            print *,i,dsqrt(w(i))/(2*pi),dsqrt(w1(i))/(2*pi)
            if(w1(i).ge.w(i))then
```

```
          write(98,*)i,dsqrt(w(i))/(2*pi),dsqrt(w1(i))/(2*pi),
     +       real(sqrt( (w1(i)-w(i))/w1(i) ))
          else
          write(98,*)i,dsqrt(w(i))/(2*pi),dsqrt(w1(i))/(2*pi),
     +       zero
          end if
          write(99,*)'mode: ',i
          write(99,*)(dpk(ii,i),ii=1,nmodes)
        else
          write(98,*)i,'       ****rigid body mode****'
        end if
       end do
       close(98)
       close(99)
       print *,'Results are in file: ',filein(1:lfile)//
     +'.eval/.evec'

      endif
      goto 999
     end if

c*************************************************************************
c*************************************************************************
c Case II.  In-fluid iterative eigensolution.  Elastic or short circuit.
c*************************************************************************
c*************************************************************************
    if(doeigen .eq.1 .and. ( ians .eq. 1)) then
c open file for output
     open(unit=99,file=filein(1:lfile)//'.eigen',status='unknown')

     print *,'choose:  [1] short circuit or [2] open circuit: '
     read *,isc

     loopcount = 0
     icase = 0
     print *,'convergence tolerance = 1 Hz'
     tol = 1.0     !  set tolerance to 1 Hz (can be changed)
     print *, 'enter in vacuo frequency'
     read(5,*) p    ! first value
     if(isc.eq.1)then
       write(99,*)'***short-circuit condition***'
     else
       write(99,*)'***open-circuit condition***'
```

```
      endif
      write(99,*)'in vacuo eigenfrequency: ',p,' Hz'
c-------------------------------------------------------------
c Beginning of the iterative loop to determine the in-water
c  eigenfrequency of interest
c-------------------------------------------------------------

c rerun chief using new frequency to generate a new zrad file.
800   write(unit=stfreq,fmt='(f10.1)') real(p)
      fchief='im'
      print *,'running CHIEF using driver: ',fchief(1:2)//
     + filein(1:lfile)
c format of system string depends on the operating system
      string=fchief(1:2)//filein(1:lfile) //'<<end\n' //
     +              stfreq(1:10)// '\nend'
      s = system(string)

      if(isc.eq.2)then
c modify stiffness for open-circuit analysis
c multiply (-1/kee) by kuevec by transpose(kuevec), add to dpk,
c   and put result in dpk
c-----------------------------------------------------------------------
c LAPACK matrix multiplication
c-----------------------------------------------------------------------
        transa='n'
        transb='t'
        m=nmodes
        k=nelecc
        n=nmodes
        alpha=-1.0/kee
        beta=1.0
        lda=nmodes
        ldb=nmodes
        ldc=nmodes
        call dgemm(transa,transb,m,n,k,alpha,kuevec, lda,
     +    kuevec,ldb,beta,dpk,ldc)
       end if

c open zrad file
      open(unit=2,file=filein(1:lfile)//'.zrad',type='old',iostat=err)
      if(err.ne.0)then
        print *,'.zrad file not found'
        stop
```

```
      end if

c-----------------------------------------------------------------------
c CHIEF results: mutual coupling matrices
c-----------------------------------------------------------------------
c read zrad files computed in CHIEF run
      read(2,*)((zr(i,j),j=1,nareac),i=1,nareac) !real
      read(2,*)((zi(i,j),j=1,nareac),i=1,nareac) !imag
      close(2)

      omega=2.0*pi*p
      omega2=omega**2

c multiply bigx by zi and put in br
c-----------------------------------------------------------------------
c LAPACK routine that does a matrix multiplication
c-----------------------------------------------------------------------
      transa='n'
      transb='n'
      alpha=1.0
      beta=0.0
      call dgemm(transa,transb,nmodes,nareac,nareac,alpha,bigx,
     +nmodes,zi,nareac,beta,br,nmodes)

c multiply (1/omega) by br by transpose(bigx), add to dpm,
c  and put result in dpm
c-----------------------------------------------------------------------
c LAPACK routine that does a matrix multiplication
c-----------------------------------------------------------------------
      transa='n'
      transb='t'
      alpha=1./omega
      beta=1.0
      call dgemm(transa,transb,nmodes,nmodes,nareac,alpha,br,
     +nmodes,bigx,nmodes,beta,dpm,nmodes)

c-----------------------------------------------------------------------
c  LAPACK Routine that computes all eigenvalues and eigenvectors
c-----------------------------------------------------------------------
      jobz='v'
      uplo='u'
      lwork=3*nmodes
      call dsygv(1, jobz, uplo, nmodes, dpk, nmodes, dpm, nmodes,
```

```
     +         w, work, lwork, info)

      if (loopcount .eq. 0) then
950     write(6,*)
       write(6,*) 'frequencies to choose from:'
       do i = case_start, case_end
        if (w(i) .gt. 0) write(6,*) i, dsqrt(w(i)) / (2*pi),
     &    ' Hz'
       end do
       write(6,*) 'choose a mode: '
       read(5,*) icase
       if (icase .lt. case_start .or. icase .gt. case_end) then
        write(6,*) 'invalid selection.  Try again.'
        goto 950
       end if
      end if

      fp = dsqrt(w(icase)) / (2.0 * pi)
      write(99,*)'estimated in-fluid frequency: ',fp,' Hz'
      loopcount = loopcount + 1
      if (abs(fp - p) .ge. tol) then
       p=fp
c-----------------------------------------------------------------
c      recover the original values of dpm and dpk.(mass and
c       stiffness matrices
c-----------------------------------------------------------------
       do i = 1, nmodes
        do j = 1, nmodes
          dpk(i,j) = dpkk(i,j)
          dpm(i,j) = dpmm(i,j)
        end do
       end do
       goto 800
      end if
      write(6,*) 'final frequency =', fp,' Hz'
      write(99,*)'final in-fluid frequency= ',fp,' Hz'
      write(99,*)'eigenvector for mode ',icase,':'
      write(99,*)(dpk(ii,icase),ii=1,nmodes)
      print *,'Results are in file: ',filein(1:lfile)//'.eigen'
     endif

c*******************************************************************************
c*******************************************************************************
```

```
c Case III - in vacuo harmonic analysis
c*************************************************************************
c*************************************************************************
     if(dohar.eq.1 .and. ( ians .eq. 0))then
       print *,'Enter applied voltage: '
       read *,phiapp
c make a copy of kuevec for later use
       do i=1,nmodes
         do j=1,nelecc
           ckuevec(i,j)=kuevec(i,j)
         end do
       end do
c open file for output of displacements
       open(unit=90,file=filein(1:lfile)//'.adat',status='unknown')
       open(unit=99,file=filein(1:lfile)//'.adisp',status='unknown')
       do freq=fr1,fr2,delfr
        print *,'freq= ',freq
        omega=2.0*pi*freq
        omega2=omega**2

c reform right hand side
       do i=1,nmodes
         do j=1,nelecc
           kuevec(i,j)=-phiapp*dreal(ckuevec(i,j))
         end do
       end do

c form dynamic stiffness matrix:  rkm = dpk - omega2*dpm
        do i=1,nmodes
          do j=1,nmodes
            rkm(i,j)=dpk(i,j)-omega2*dpm(i,j)
          end do
        end do

c---------------------------------------------------------------------
c LAPACK routine  that uses a LU factorization to compute the
c  solution to a real complex system of linear equations
c---------------------------------------------------------------------
        fact='n'
        equed='n'
        trans='n'
        call dgesvx(fact,trans,nmodes,nelecc,rkm,nmodes,af,nmodes,
   +    ipiv,equed,r,c,kuevec,nmodes,x,nmodes,rcond,ferr,berr,work,
```

```
     +    iwork,info)

          write(99,*)'displacements for frequency: ',freq
          write(99,*)x

c solve for input admittance
c get kuevec back
         do i=1,nmodes
           do j=1,nelecc
             kuevec(i,j)=dreal(ckuevec(i,j))
           end do
         end do

c multiply transpose(kuevec)by x and put in phir
c-----------------------------------------------------------------------
c LAPACK routine that does a matrix multiplication
c-----------------------------------------------------------------------
          transa='t'
          transb='n'
          m=nelecc
          k=nmodes
          n=nelecc
          alpha=1.0
          beta=0.0
          lda=nmodes
          ldb=nmodes
          ldc=nelecc
          call dgemm(transa,transb,m,n,k,alpha,kuevec,lda,
     +    x,ldb,beta,phir,ldc)
c add kee*phiapp to phir, multiply by omega, and divide by phiapp
          phir(1,1)=-(phir(1,1)+kee*phiapp)*omega/phiapp
          write(99,*)'input admittance= j*',phir(1,1)
          write(90,*)freq,x(20,1),phir(1,1)
        enddo
       print *,'Results are in file: ',filein(1:lfile)//'.adisp'
       close(90)
       close(99)
       goto 999
      endif

c*********************************************************************
c*********************************************************************
c Case IV - in fluid harmonic radiation analysis
```

```
c*************************************************************************
c*************************************************************************
     if((dohar.eq.1).and.(ians.eq.1).and.(irad.eq.1))then
      print *,'Enter applied voltage: '
      read *,phiapp
      err=0
      open(unit=91,file=filein(1:lfile)//'.fdat',status='unknown')
      open(unit=99,file=filein(1:lfile)//'.fdisp',status='unknown')
      do freq=fr1,fr2,delfr

c form/reform right hand side
        do i=1,nmodes
         do j=1,nelecc
           ckuevec(i,j)=-phiapp*kuevec(i,j)
         end do
        end do

c run CHIEF
        write(unit=stfreq,fmt='(f10.1)') freq
        fchief='im'
        print *,'running CHIEF driver: ',fchief(1:2)//filein(1:lfile)

c system command goes out and runs the CHIEF driver program
c format of system string depends on operating system
        string=fchief(1:2)//filein(1:lfile) //'<<end\n' //
     +              stfreq(1:10)// '\nend'
        s = system(string)
        open(unit=2,file=filein(1:lfile)//'.zrad',type='old',
     +       iostat=err)
        if(err.ne.0)then
          print *,'.zrad file not found'
          stop
        endif
c-----------------------------------------------------------------
c  CHIEF output: mutual coupling matrices
c-----------------------------------------------------------------
c read zrad
        read(2,*)((zr(i,j),j=1,nareac),i=1,nareac) ! real
        read(2,*)((zi(i,j),j=1,nareac),i=1,nareac) ! imaginary
        close(2)
        do i=1,nareac
          do j=1,nareac
            zrad(i,j)=dcmplx(zr(i,j),zi(i,j))
```

```
        enddo
       enddo
       omega=2.0*pi*freq
       omega2=omega**2

c form dynamic stiffness matrix:  crkm = dpk - omega2*dpm
       do i=1,nmodes
        do j=1,nmodes
        crkm(i,j)=dpk(i,j)-omega2*dpm(i,j)
        end do
       end do

c check compatibility matrix: each column should sum to +/-1 because of unit pressure
       if(freq.eq.fr1)then
        print *,'checking compatibility matrix'
        do j=1,nareac
         sum=0
         do i=1,nmodes
          sum=sum+cbigx(i,j)
         enddo
         if((abs(sum)-1.).gt.0.01)then
           print *,'Sum of column ',j,' is not unity; sum=  ',sum
           print *,'STOP AND CHECK THE COMPATIBILITY MATRIX!!'
           goto 999
         end if
        enddo
       endif

c multiply omega by cbigx by zrad and put in b
c -------------------------------------------------------------------
c  LAPACK complex matrix multiplication
c-------------------------------------------------------------------
       transa='n'
       transb='n'
       xalpha=dcmplx(0.d0,omega)
       xbeta=dcmplx(0.d0,0.d0)
       call zgemm(transa,transb,nmodes,nareac,nareac,xalpha,cbigx,
     +  nmodes,zrad,nareac,xbeta,b,nmodes)

c multiply b by transpose(cbigx), add to crkm, and put result in crkm
c -------------------------------------------------------------------
c  LAPACK complex matrix multiplication
c-------------------------------------------------------------------
```

```
      transa='n'
      transb='t'
      xalpha=dcmplx(1.,0.)
      xbeta=dcmplx(1.,0.)
      call zgemm(transa,transb,nmodes,nmodes,nareac,xalpha,b,
     +  nmodes,cbigx,nmodes,xbeta,crkm,nmodes)

c solve (crkm)(cx)=ckuevec for cx
c-----------------------------------------------------------------------
c LAPACK routine  that uses a LU factorization to compute the
c  solution to a complex system of linear equations
c-----------------------------------------------------------------------
      fact='n'
      equed='n'
      trans='n'
      call zgesvx(fact,trans,nmodes,nelecc,crkm,nmodes,aff,nmodes,
     +  ipiv,equed,r,c,ckuevec,nmodes,cx,nmodes,rcond,ferr,berr,
     +  cwork,rwork,info)
      write(99,*)'displacements for frequency: ',freq
      write(99,*)cx

c get back ckuevec since it was overwritten by the solver
     do i=1,nmodes
       do j=1,nelecc
         ckuevec(i,j)=kuevec(i,j)
       end do
     end do

c solve for input admittance
c multiply transpose(ckuevec) by cx and put in phi
c-----------------------------------------------------------------------
c LAPACK routine that does a matrix multiplication
c-----------------------------------------------------------------------
      transa='t'
      transb='n'
      m=nelecc
      k=nmodes
      n=nelecc
      xalpha=1.0
      xbeta=0.0
      lda=nmodes
      ldb=nmodes
      ldc=nelecc
```

```
        call zgemm(transa,transb,m,n,k,xalpha,ckuevec,lda,
     +   cx,ldb,xbeta,phi,ldc)
c add kee*phiapp to phi, multiply by j*omega, and divide by phiapp
        phi(1,1)=-(phi(1,1)+kee*phiapp)*cmplx(0.d0,omega)/phiapp
        write(99,*)'input admittance= ',phi(1,1)

c solve for normal surface velocities
c multiply (j*omega) by transpose(cbigx) by cx and put in pinc
c ---------------------------------------------------------------
c  LAPACK complex matrix multiplication
c---------------------------------------------------------------
        transa='t'
        transb='n'
        xalpha=dcmplx(0.d0,omega)
        xbeta=dcmplx(0.,0.)
        call zgemm(transa,transb,nareac,1,nmodes,xalpha,cbigx,
     +   nmodes,cx,nmodes,xbeta,pinc,nareac)
       write(99,*)'normal surface velocities: '
       write(99,*)pinc
       write(91,*)freq,real(cdabs(cx(20,1))),real(cdabs(phi(1,1))),
     &   real(cdabs(pinc(4,1)))
      enddo
      print *,'Results are in file: ',filein(1:lfile)//'.fdisp'
      close(91)
      close(99)
      goto 999
     endif

c*************************************************************************
c*************************************************************************
c Case V - in fluid harmonic scattering analysis
c*************************************************************************
c*************************************************************************
     if((dohar.eq.1).and.(ians.eq.1).and.(irad.eq.0))then
      err=0

c open file for output
      open(unit=92,file=filein(1:lfile)//'.fpdat',status='unknown')
      open(unit=99,file=filein(1:lfile)//'.fpdisp',status='unknown')

c modify stiffness for open-circuit analysis
c multiply (-1/kee) by kuevec by transpose(kuevec), add to dpk,
c   and put result in dpk
```

```
c-----------------------------------------------------------------------
c LAPACK matrix multiplication
c-----------------------------------------------------------------------
      transa='n'
      transb='t'
      m=nmodes
      k=nelecc
      n=nmodes
      alpha=-1.0/kee
      beta=1.0
      lda=nmodes
      ldb=nmodes
      ldc=nmodes
      call dgemm(transa,transb,m,n,k,alpha,kuevec, lda,
     +    kuevec,ldb,beta,dpk,ldc)

c check compatibility matrix: each column should sum to +/-1 because of unit pressure
      print *,'checking compatibility matrix'
      do j=1,nareac
       sum=0
       do i=1,nmodes
        sum=sum+cbigx(i,j)
       enddo
       if((abs(sum)-1.).gt.0.01)then
         print *,'Sum of column  ',j,' is not unity; sum=  ',sum
         print *,'STOP AND CHECK THE COMPATIBILITY MATRIX!!'
       end if
      enddo

c frequency loop
      do freq=fr1,fr2,delfr

c run CHIEF
       write(unit=stfreq,fmt='(f10.1)') freq
       fchief='sc1'
       print *,'running CHIEF driver program: ',fchief(1:3)//
     +  filein(1:lfile)

c system command goes out and runs the CHIEF driver program
c format of system string depends on operating system
       string=fchief(1:3)//filein(1:lfile) //'<<end\n' //
     +              stfreq(1:10)// '\nend'
       s = system(string)
```

```
c-----------------------------------------------------------------
c  CHIEF output: mutual coupling matrices
c-----------------------------------------------------------------
      open(unit=2,file=filein(1:lfile)//'.zrad',type='old',
     +     iostat=err)
      if(err.ne.0)then
        print *,'.zrad file not found'
        stop
      endif
      read(2,*)((zr(i,j),j=1,nareac),i=1,nareac) !real
      read(2,*)((zi(i,j),j=1,nareac),i=1,nareac) !imag
      close(2)
      do i=1,nareac
        do j=1,nareac
         zrad(i,j)=dcmplx(zr(i,j),zi(i,j))
        enddo
      enddo

c-----------------------------------------------------------------------------
c Incident pressure on each surface
c-----------------------------------------------------------------------------
      open(unit=2,file=filein(1:lfile)//'.pinc',type='old',
     +     iostat=err)
      if(err.ne.0)then
        print *,'.pinc file not found'
        stop
      endif
      do i=1,nareac
        read(2,*)pincr,pinci
        pinc(i,1)=dcmplx(pincr,pinci)
      end do
      close(2)

c set up complex dynamic stiffness matrix:  crkm = dpk - omega2*dpm
      omega=2.0*pi*freq
      omega2=omega**2
      do i=1,nmodes
        do j=1,nmodes
        crkm(i,j)=dpk(i,j)-omega2*dpm(i,j)
        end do
      end do

c forming part of the left hand side of the equation
```

```
c multiply (j*omega) by cbigx by zrad, and put in b
c -----------------------------------------------------------------
c  LAPACK complex matrix multiplication
c-----------------------------------------------------------------
      transa='n'
      transb='n'
      xalpha=dcmplx(0.d0,omega)
      xbeta=dcmplx(0.d0,0.d0)
      call zgemm(transa,transb,nmodes,nareac,nareac,xalpha,cbigx,
     +    nmodes,zrad,nareac,xbeta,b,nmodes)

c forming the left hand side of the equation to be solved
c multiply b by transpose(cbigx), add to crkm, and put result in crkm
c -----------------------------------------------------------------
c  LAPACK complex matrix multiplication
c-----------------------------------------------------------------
      transa='n'
      transb='t'
      xalpha=dcmplx(1.,0.)
      xbeta=dcmplx(1.,0.)
      call zgemm(transa,transb,nmodes,nmodes,nareac,xalpha,b,
     +    nmodes,cbigx,nmodes,xbeta,crkm,nmodes)

c forming the right hand side of the equation to be solved
c multiply -1 by cbigx by pinc and put in ckuevec
c -----------------------------------------------------------------
c  LAPACK complex matrix multiplication
c-----------------------------------------------------------------
      transa='n'
      transb='n'
      xalpha=dcmplx(-1.,0.)
      xbeta=dcmplx(0.,0.)
      call zgemm(transa,transb,nmodes,nelecs,nareac,xalpha,cbigx,
     +    nmodes,pinc,nareac,xbeta,ckuevec,nmodes)

c solve for displacements
c solve (crkm)(cx)=ckuevec for cx
c-----------------------------------------------------------------
c LAPACK routine  that uses a LU factorization to compute the
c  solution to a  complex system of linear equations
c-----------------------------------------------------------------
      fact='n'
      equed='n'
```

```
      trans='n'
      call zgesvx(fact,trans,nmodes,nelecs,crkm,nmodes,aff,nmodes,
     +  ipiv,equed,r,c,ckuevec,nmodes,cx,nmodes,rcond,ferr,berr,
     +  cwork,rwork,info)
      write(99,*)'displacements for frequency: ',freq
      write(99,*)cx

c solve for normal surface velocities
c multiply (j*omega) by transpose(cbigx) by cx and put in pinc
c ---------------------------------------------------------------
c  LAPACK complex matrix multiplication
c---------------------------------------------------------------
      transa='t'
      transb='n'
      xalpha=dcmplx(0.d0,omega)
      xbeta=dcmplx(0.,0.)
      call zgemm(transa,transb,nareac,1,nmodes,xalpha,cbigx,
     +  nmodes,cx,nmodes,xbeta,pinc,nareac)
      write(99,*)'normal surface velocities: '
      write(99,*)pinc

c need to put piezoelectric stiffness back in ckuevec
       do i=1,nmodes
         do j=1,nelecc
           ckuevec(i,j)=dcmplx(kuevec(i,j),0.d+0)
         enddo
       enddo

c solve for external potential
c multiply (-1/kee) by transpose(ckuevec) by cx and put in phi
c---------------------------------------------------------------
c LAPACK routine that does a matrix multiplication
c---------------------------------------------------------------
       transa='t'
       transb='n'
       m=nelecc
       k=nmodes
       n=nelecc
       xalpha=-1.0/kee
       xbeta=0.0
       lda=nmodes
       ldb=nmodes
       ldc=nelecc
```

```
      call zgemm(transa,transb,m,n,k,xalpha,ckuevec,lda,
     +   cx,ldb,xbeta,phi,ldc)
      write(99,*)'phi= ',phi(1,1)
     write(92,*)freq,real(cdabs(cx(20,1))),real(cdabs(phi(1,1))),
     &  real(cdabs(pinc(4,1)))
     enddo
     print *,'Results are in file: ',filein(1:lfile)//'.fpdisp'
     close(92)
     close(99)
     goto 999
    endif
999  continue
c*******************************************************************************
c*******************************************************************************
9999 end
```

```
*************************************************************************
```

CHIEF Driver Program: imtest1.for

```
*************************************************************************
C   Program: imtest1.for
C
C   ***** CONTROL 88 *****

C   PROGRAM CHIEF88 DRIVER
C   MXSREG - MAXIMUM NUMBER OF SURFACE REGIONS
C   MXIPS  - MAXIMUM NUMBER OF INTERIOR POINTS
C   MXARS  - MAXIMUM NUMBER OF SURFACE SUBDIVISIONS/SYM BLK
C   MXGAUS - MAXIMUM ORDER OF GAUSSIAN QUADRATURE
C   MXQPTS - MAXIMUM NUMBER OF QUADRATURE POINTS/SUBDIVISION
C   MXBLKS - MAXIMUM NUMBER OF SYMMETRY BLOCKS
C   MXFFP  - MAXIMUM NUMBER OF FAR-FIELD POINTS
C   MXNFP  - MAXIMUM NUMBER OF NEAR-FIELD POINTS
C   MXPTSC - MAXIMUM NUMBER OF POINT SOURCES
C   MAXCOR - MAXIMUM NUMBER OF FINITE ELEMENT NODES
C   MXFPS - MAXO(MXARS+MXIPS,MXFFP,MXNFP)

    PARAMETER (MXSREG=1200)
    PARAMETER (MAXCOR=5000)
    PARAMETER (MXIPS=20)
    PARAMETER (MXARS=1200)
    PARAMETER (MXGAUS=64)
    PARAMETER (MXQPTS=512)
    PARAMETER (MXBLKS=100)
    PARAMETER (MXFFP=361)
    PARAMETER (MXNFP=361)
    PARAMETER (MXPTSC=20)
    PARAMETER (MXFPS=1250)
C   PARAMETER (MXFPS=MAX0(MXARS+MXIPS,MXFFP,MXNFP))
C   PARAMETER (NWDVEC=2*MXARS)
    PARAMETER (NBLKS = 8)
C
C Input commons
C
C Thin-body and mixed-body commons.  (T. W. Wu)
C
    COMMON /THIN/ ITHIN,IFLAT,NDTHIN,NDEXPD
    COMMON /NORMAL/ PN(4,MXFPS),PNN(4,MXFPS)
    COMMON /MIX/ IMIX,BODY(MXSREG),IBODY(MXARS)
    COMMON /MIX2/ NDSUMP,NDSUMV,NRE,NTH
    CHARACTER*1 BODY
    COMMON /CONST/  RHO,C
    COMMON /EPSLON/ ALT(3,3,3)
    COMMON /PRTCOM/ NUNPRT,NUNERR
    COMMON /PRTRD/  RUNID,DATE
    CHARACTER*32 RUNID
    CHARACTER*8  DATE
    COMMON /NDASG/  NDQPTS,NDPMXS,NDVMXS,NDDECM,NDVELS,NDSPS,
   1          NDPMXF,NDVMXF,NDPMXN,NDVMXN,NDPSSP,NDEXPR,NDCOMV,
```

```
     1          NDTEMP,NDZRDB,NDPATB,NTMPVEL,NTEMP1,NTEMP2,NTEMP3
      COMMON /SVALS/  NSREG,NSEQNS(MXSREG),SUL(MXSREG),SUU(MXSREG),
     1          SVL(MXSREG),SVU(MXSREG),NSU(MXSREG),NSV(MXSREG),
     1          CCS(10,MXSREG),TRNSS(3,MXSREG),IZAX(MXSREG),
     1          IORDU(MXSREG),IORDV(MXSREG),NCCEQS
      COMMON /CORD/   COORDS(MAXCOR,3)
      COMMON /IPTS/   NUMIPS,IPXS(3,MXIPS)
      REAL IPXS
      COMMON /PTSINP/ NUMPTS,PTSRCS(4,MXPTSC),PTSWT(MXPTSC),
     1          IOPTSC(MXPTSC)
      COMPLEX PTSWT
      COMMON /PLWINP/ AINC,THTINC,PHIINC,ISCATR
      COMMON /BAFFLE/ INFBAF
      COMMON /FFINP/  NUMTHP,THTPHI(2,MXFFP)
      COMMON /NFINP/  NUMFPN,NFPXS(3,MXNFP)
      REAL NFPXS
C
C Output commons
C
      COMMON /TAPREC/ RECRD(10),IRECRD(30)
      COMMON /TAPRC1/ ARECRD(10)
      CHARACTER*8 ARECRD
      COMMON /PRGVLS/ NDIMPV,NUMARS,NUMSFP,NUMFFP,NUMNFP,NWDVEC
      COMMON /SURARS/ AREAS(MXARS)
      COMMON /ODSVEC/ TVECT(MXARS),B(MXARS),IPIVTR(MXFPS)
      COMPLEX TVECT, B
      COMMON /VELSPS/ VEL(MXARS),SP(MXARS)
      COMPLEX VEL,SP
      COMMON /PDISL/  POWER,DIRIND,SRCRVL
      COMMON /FFVALS/ FFP(MXFFP),PNRMFF(MXFFP),IFFNRM,RMFNRM
      COMPLEX FFP
      COMMON /TSCOM/  TGTSTH(MXFFP)
      COMMON /NFVALS/ NFP(MXNFP),PNRMNF(MXNFP),INFNRM,RMNNRM
      COMPLEX NFP
      COMMON /PTSCOM/ PTSSP(MXARS)
      COMPLEX PTSSP
      COMMON /EXTCOM/ EXTPRS(MXFPS),IEXTFG
      COMPLEX EXTPRS
      COMMON /NBPRTC/ IRHSPT,NARSPT,NPTBLK,FRQPT
      COMMON /NBPRTS/ SYMTPT
      CHARACTER*3 SYMTPT

C*************** impedance coating modifications ***************
      COMMON /COATING/ ZCOAT, NUMZ
      COMPLEX ZCOAT(MXARS)
C NUMZ - total number of impedance layer surfaces (NSU * NSV)
C*************************************************************

      DIMENSION CC(10), TRNS(3), IELTS(8, 300)
      REAL X1(1000), Y1(1000)

      CHARACTER*3 SYMTYP
```

```
      CHARACTER*4 FLDTYP, TAPEID, PRTTYP
      INTEGER   XIZAX, XNSEQNS, XIRG, XNSU, XNSV, XIORDU, XIORDV
      INTEGER   ICOOR, IELEM
      REAL      THETAA(MXFFP)
      COMPLEX   FFPP(MXFFP)
      COMPLEX   PMATX(16), VMATX(16)
      COMPLEX   ZMTX(4, 4)
      DIMENSION ZRI(2, 4, 4)
      EQUIVALENCE (ZMTX(1,1), ZRI(1,1,1))

      MDSIZE = 16
      RUNID  = 'test1'
      DATE   = '09/07/95'

      CALL INITCM
      CALL OPNSFL

      RHO    = 1000.00
      C      = 1500.00
      NUMZ   = 0
      NUMARS = 4
      OPEN(UNIT=NUNPRT,FILE='test1.out',STATUS='UNKNOWN',
     1   FORM='FORMATTED')


C Symmetry Inputs

      PI = ACOS(-1.0)

      SYMTYP  = 'ROT'
      IRHSYM  = 1
C Surface Region Inputs.

      ROTLIM = PI / NBLKS
      DEG2RAD = PI / 180.0
      XIRG   = 0

      DO 10 I = 1, 10
        CC(I) = 0.0
10    CONTINUE

      DO 20 I = 1, 3
        TRNS(I) = 0.0
20    CONTINUE

C Define region:  rod surfaces (Linear or Quadradic Axisymmetric Interpolation)

      XNSEQNS = 11

C Read in coordinate and element data.

      ICOOR = 0
```

```
      OPEN(UNIT=8,FILE='iomagc.test1',STATUS='OLD',FORM='FORMATTED')
30    READ(8,*,ERR=50) IDUM, (COORDS(IDUM,J),J=1,2)
      IF (IDUM .GT. ICOOR) ICOOR = IDUM
      IF (IDUM .GT. MAXCOR) THEN
        WRITE (6,*) 'Error:  Coordinate index in coordinate ' //
     +          'data is greater than MAXCOR.'
        STOP
      END IF
      GOTO 30
50    CLOSE(8)

C Read in element data form external file.

      IELEM = 0
      OPEN(UNIT=8,FILE='ioelms.test1',STATUS='OLD',FORM='FORMATTED')
60    READ(8,*,ERR=80) IDUM, (IELTS(J, IELEM+1), J=1,3)
      IELEM = IELEM + 1
      GOTO 60
80    CLOSE(8)

C Display coordinate data.

      WRITE(NUNPRT,90)
90    FORMAT(1H1,'COORDINATE DATA',/)
      DO 110 I = 1, ICOOR
        WRITE(NUNPRT,100) I, (COORDS(I,J), J=1, 2)
100     FORMAT(I5,5X,2E15.4)
110   CONTINUE

C Display element data.

      WRITE(NUNPRT,120)
120   FORMAT(//,' ELEMENT DATA',/)
      DO 140 I = 1, IELEM
        WRITE(NUNPRT,130) I, (IELTS(J,I),J=1,8)
130     FORMAT(I5,10X,8I5)
140   CONTINUE

      TRNS(1) = 0.00000
      TRNS(2) = 0.00000
      TRNS(3) = 0.00000
      XIZAX  = 3
      XSUL   = -1.0
      XSUU   =  1.0
      XSVL   = -ROTLIM
      XSVU   =  ROTLIM
      XNSU   = 1
      XNSV   = 1
      XIORDU = 8
      XIORDV = 8

      DO 150 I = 1, IELEM
```

```
      DO 160 J = 1, 3
        CC(J) = IELTS(J,I)
160     CONTINUE
      XIRG = XIRG + 1
      CALL LDSURR(XIRG, XNSEQNS, CC, TRNS, XIZAX,
     1          XSUL, XSUU, XSVL, XSVU, XNSU, XNSV,
     1          XIORDU, XIORDV)
150   CONTINUE


    NSREG = XIRG

C The syntax for the PLOTCHIEF call is as follows:
C
C CALL PLOTCHIEF(CHARACTER*(*) RUNID,   ! title of CHIEF run.
C           INTEGER     NBLKS,   ! number of symmetry blocks used.
C           CHARACTER*3  SYMTYP,  ! symmetry type from CHIEF.
C           INTEGER     ISUBDIV,  ! 0 - uses optimum subdivisions.
C                       ! 1 - uses NSU's and NSV's defined in LDSURR.
C           REAL        AX,AY,AZ, ! rotation in degrees about the
C                       ! X, Y, and Z axes.
C
C For example:
C     CALL PLOTCHIEF(RUNID, NBLKS, SYMTYP, 1, 30.0, 60.0, 0.0, 0.0, 0.0)

    CALL PRTOUT('GEOM', 1)

    OPEN(UNIT=2,FILE='test1.zrad',STATUS='UNKNOWN')

C Define Frequency Loop.

    DO 180 FREQ = 14511.0, 14516.0, 1.00000

C Generate radiation impedance matricies.

C Generate Surface P and V Matrices.
    CALL SURMAT(FREQ, SYMTYP, NBLKS, PMATX, VMATX, MDSIZE)

    IRHSYM = 1

C Decompose Matrices
    CALL DECOMM(SYMTYP, NBLKS, IRHSYM, PMATX, VMATX, MDSIZE)

    NUMZ = 0
    CALL ZRADMX(FREQ, ZMTX, NUMARS, SYMTYP, NBLKS,
    1         PMATX, VMATX, MDSIZE)

    NUMZ = 0
    WRITE(2,*) ((ZRI(1,I,J)*NBLKS/2.0/PI, J=1,NUMARS-NUMZ),
    &           I=1,NUMARS-NUMZ)
    WRITE(2,*) ((ZRI(2,I,J)*NBLKS/2.0/PI, J=1,NUMARS-NUMZ),
    &           I=1,NUMARS-NUMZ)
```

```
180  CONTINUE
     STOP
     END
```

```
************************************************************************
CHIEF Driver Program: tvtest1.for
************************************************************************
C   Program: tvtest1.for
C
C   ***** CONTROL 88 *****

C   PROGRAM CHIEF88 DRIVER
C   MXSREG - MAXIMUM NUMBER OF SURFACE REGIONS
C   MXIPS  - MAXIMUM NUMBER OF INTERIOR POINTS
C   MXARS  - MAXIMUM NUMBER OF SURFACE SUBDIVISIONS/SYM BLK
C   MXGAUS - MAXIMUM ORDER OF GAUSSIAN QUADRATURE
C   MXQPTS - MAXIMUM NUMBER OF QUADRATURE POINTS/SUBDIVISION
C   MXBLKS - MAXIMUM NUMBER OF SYMMETRY BLOCKS
C   MXFFP  - MAXIMUM NUMBER OF FAR-FIELD POINTS
C   MXNFP  - MAXIMUM NUMBER OF NEAR-FIELD POINTS
C   MXPTSC - MAXIMUM NUMBER OF POINT SOURCES
C   MAXCOR - MAXIMUM NUMBER OF FINITE ELEMENT NODES
C   MXFPS - MAXO(MXARS+MXIPS,MXFFP,MXNFP)

    PARAMETER (MXSREG=1200)
    PARAMETER (MAXCOR=5000)
    PARAMETER (MXIPS=20)
    PARAMETER (MXARS=1200)
    PARAMETER (MXGAUS=64)
    PARAMETER (MXQPTS=512)
    PARAMETER (MXBLKS=100)
    PARAMETER (MXFFP=361)
    PARAMETER (MXNFP=361)
    PARAMETER (MXPTSC=20)
    PARAMETER (MXFPS=1250)
C   PARAMETER (MXFPS=MAX0(MXARS+MXIPS,MXFFP,MXNFP))
C   PARAMETER (NWDVEC=2*MXARS)
    PARAMETER (NBLKS = 8)
C
C Input commons
C
C Thin-body and mixed-body commons.  (T. W. Wu)
C
    COMMON /THIN/ ITHIN,IFLAT,NDTHIN,NDEXPD
    COMMON /NORMAL/ PN(4,MXFPS),PNN(4,MXFPS)
    COMMON /MIX/ IMIX,BODY(MXSREG),IBODY(MXARS)
    COMMON /MIX2/ NDSUMP,NDSUMV,NRE,NTH
    CHARACTER*1 BODY
    COMMON /CONST/  RHO,C
    COMMON /EPSLON/ ALT(3,3,3)
    COMMON /PRTCOM/ NUNPRT,NUNERR
    COMMON /PRTRD/  RUNID,DATE
    CHARACTER*32 RUNID
    CHARACTER*8  DATE
    COMMON /NDASG/  NDQPTS,NDPMXS,NDVMXS,NDDECM,NDVELS,NDSPS,
   1          NDPMXF,NDVMXF,NDPMXN,NDVMXN,NDPSSP,NDEXPR,NDCOMV,
```

```
     1          NDTEMP,NDZRDB,NDPATB,NTMPVEL,NTEMP1,NTEMP2,NTEMP3
     COMMON /SVALS/  NSREG,NSEQNS(MXSREG),SUL(MXSREG),SUU(MXSREG),
     1          SVL(MXSREG),SVU(MXSREG),NSU(MXSREG),NSV(MXSREG),
     1          CCS(10,MXSREG),TRNSS(3,MXSREG),IZAX(MXSREG),
     1          IORDU(MXSREG),IORDV(MXSREG),NCCEQS
     COMMON /CORD/  COORDS(MAXCOR,3)
     COMMON /IPTS/  NUMIPS,IPXS(3,MXIPS)
     REAL IPXS
     COMMON /PTSINP/ NUMPTS,PTSRCS(4,MXPTSC),PTSWT(MXPTSC),
     1          IOPTSC(MXPTSC)
     COMPLEX PTSWT
     COMMON /PLWINP/ AINC,THTINC,PHIINC,ISCATR
     COMMON /BAFFLE/ INFBAF
     COMMON /FFINP/  NUMTHP,THTPHI(2,MXFFP)
     COMMON /NFINP/  NUMFPN,NFPXS(3,MXNFP)
     REAL NFPXS
C
C Output commons
C
     COMMON /TAPREC/ RECRD(10),IRECRD(30)
     COMMON /TAPRC1/ ARECRD(10)
     CHARACTER*8 ARECRD
     COMMON /PRGVLS/ NDIMPV,NUMARS,NUMSFP,NUMFFP,NUMNFP,NWDVEC
     COMMON /SURARS/ AREAS(MXARS)
     COMMON /ODSVEC/ TVECT(MXARS),B(MXARS),IPIVTR(MXFPS)
     COMPLEX TVECT, B
     COMMON /VELSPS/ VEL(MXARS),SP(MXARS)
     COMPLEX VEL,SP
     COMMON /PDISL/  POWER,DIRIND,SRCRVL
     COMMON /FFVALS/ FFP(MXFFP),PNRMFF(MXFFP),IFFNRM,RMFNRM
     COMPLEX FFP
     COMMON /TSCOM/  TGTSTH(MXFFP)
     COMMON /NFVALS/ NFP(MXNFP),PNRMNF(MXNFP),INFNRM,RMNNRM
     COMPLEX NFP
     COMMON /PTSCOM/ PTSSP(MXARS)
     COMPLEX PTSSP
     COMMON /EXTCOM/ EXTPRS(MXFPS),IEXTFG
     COMPLEX EXTPRS
     COMMON /NBPRTC/ IRHSPT,NARSPT,NPTBLK,FRQPT
     COMMON /NBPRTS/ SYMTPT
     CHARACTER*3 SYMTPT

C*************** impedance coating modifications ***************
     COMMON /COATING/ ZCOAT, NUMZ
     COMPLEX ZCOAT(MXARS)
C  NUMZ - total number of impedance layer surfaces (NSU * NSV)
C*************************************************************

     DIMENSION CC(10), TRNS(3), IELTS(8, 300)
     REAL X1(1000), Y1(1000)

     CHARACTER*3 SYMTYP
```

```
      CHARACTER*4 FLDTYP, TAPEID, PRTTYP
      INTEGER    XIZAX, XNSEQNS, XIRG, XNSU, XNSV, XIORDU, XIORDV
      INTEGER    ICOOR, IELEM
      REAL       THETAA(MXFFP)
      COMPLEX    FFPP(MXFFP)
      COMPLEX   PMATX(2812), VMATX(2812)
      MDSIZE = 2812
      RUNID  = 'test1'
      DATE   = '09/07/95'

      CALL INITCM
      CALL OPNSFL

      RHO    = 1000.00
      C      = 1500.00
      NUMZ   = 0
      NUMARS = 4
      OPEN(UNIT=NUNPRT,FILE='test1.out',STATUS='UNKNOWN',
     1    FORM='FORMATTED')


C Symmetry Inputs

      PI = ACOS(-1.0)

      SYMTYP  = 'ROT'
      IRHSYM  = 1
C Surface Region Inputs.

      ROTLIM = PI / NBLKS
      DEG2RAD = PI / 180.0
      XIRG   = 0

      DO 10 I = 1, 10
        CC(I) = 0.0
10    CONTINUE

      DO 20 I = 1, 3
        TRNS(I) = 0.0
20    CONTINUE

C Define region:  rod surfaces (Linear or Quadradic Axisymmetric Interpolation)

      XNSEQNS = 11

C  Read in coordinate and element data.

      ICOOR = 0
      OPEN(UNIT=8,FILE='iomagc.test1',STATUS='OLD',FORM='FORMATTED')
30    READ(8,*,ERR=50) IDUM, (COORDS(IDUM,J),J=1,2)
      IF (IDUM .GT. ICOOR) ICOOR = IDUM
      IF (IDUM .GT. MAXCOR) THEN
```

```
      WRITE (6,*) 'Error:  Coordinate index in coordinate ' //
     +           'data is greater than MAXCOR.'
      STOP
    END IF
    GOTO 30
50  CLOSE(8)

C Read in element data form external file.

    IELEM = 0
    OPEN(UNIT=8,FILE='ioelms.test1',STATUS='OLD',FORM='FORMATTED')
60  READ(8,*,ERR=80) IDUM, (IELTS(J, IELEM+1), J=1,3)
    IELEM = IELEM + 1
    GOTO 60
80  CLOSE(8)

C Display coordinate data.

    WRITE(NUNPRT,90)
90  FORMAT(1H1,'COORDINATE DATA',/)
    DO 110 I = 1, ICOOR
      WRITE(NUNPRT,100) I, (COORDS(I,J), J=1, 2)
100   FORMAT(I5,5X,2E15.4)
110 CONTINUE

C Display element data.

    WRITE(NUNPRT,120)
120 FORMAT(///,' ELEMENT DATA',/)
    DO 140 I = 1, IELEM
      WRITE(NUNPRT,130) I, (IELTS(J,I),J=1,8)
130   FORMAT(I5,10X,8I5)
140 CONTINUE

    TRNS(1) = 0.00000
    TRNS(2) = 0.00000
    TRNS(3) = 0.00000
    XIZAX  = 3
    XSUL   = -1.0
    XSUU   =  1.0
    XSVL   = -ROTLIM
    XSVU   =  ROTLIM
    XNSU   = 1
    XNSV   = 1
    XIORDU = 8
    XIORDV = 8

    DO 150 I = 1, IELEM
      DO 160 J = 1, 3
        CC(J) = IELTS(J,I)
160   CONTINUE
      XIRG = XIRG + 1
```

```
      CALL LDSURR(XIRG, XNSEQNS, CC, TRNS, XIZAX,
     1          XSUL, XSUU, XSVL, XSVU, XNSU, XNSV,
     1          XIORDU, XIORDV)
  150 CONTINUE


      NSREG = XIRG

C The syntax for the PLOTCHIEF call is as follows:
C
C CALL PLOTCHIEF(CHARACTER*(*) RUNID,   ! title of CHIEF run.
C           INTEGER     NBLKS,   ! number of symmetry blocks used.
C           CHARACTER*3  SYMTYP,  ! symmetry type from CHIEF.
C           INTEGER     ISUBDIV,  ! 0 - uses optimum subdivisions.
C                       ! 1 - uses NSU's and NSV's defined in LDSURR.
C           REAL       AX,AY,AZ, ! rotation in degrees about the
C                       ! X, Y, and Z axes.
C
C For example:
C     CALL PLOTCHIEF(RUNID, NBLKS, SYMTYP, 1, 30.0, 60.0, 0.0, 0.0, 0.0)

      CALL PRTOUT('GEOM', 1)

      OPEN(UNIT=3,FILE='test1.ffpb',STATUS='UNKNOWN')
      OPEN(UNIT=5,FILE='test1.patt',STATUS='UNKNOWN')
      OPEN(UNIT=4,FILE='test1.vel',STATUS='OLD')

C Define Frequency Loop.

      DO 180 FREQ = 14511.0, 14516.0, 1.00000

C Generate surface P and V matrices.

      CALL SURMAT(FREQ, SYMTYP, NBLKS, PMATX, VMATX, MDSIZE)

C Decompose Matrices

      CALL DECOMM(SYMTYP, NBLKS, IRHSYM, PMATX, VMATX, MDSIZE)

      CALL IOSUB(NDVELS, 10, VEL, 0)

      DO 190 I = 1, 4
        READ(4,*) VEL(I)
  190 CONTINUE

      CALL IOSUB(NDVELS, 1, VEL, NWDVEC)

C Generate Surface Pressures.

      CALL SURPRS(FREQ, SYMTYP, NBLKS, IRHSYM, PMATX, VMATX, MDSIZE)

      CALL PRTOUT('VEL', 1)
```

```
      CALL PRTOUT('SP', 1)

      FLDTYP = 'FAR'

      NUMTHP = 703
      ICOUNT = 0
      DO 210 I = 0, 180, 10
        DO 200 J = 0, 360, 10
          ICOUNT = ICOUNT + 1
          THTPHI(1,ICOUNT) = J
          THTPHI(2,ICOUNT) = I
200     CONTINUE
210   CONTINUE

C Calculate Far-Field Matrices.
      CALL FLDMAT(FREQ, SYMTYP, FLDTYP, NBLKS, PMATX, VMATX, MDSIZE)

      IFFNRM = 37

C Calculate Far-Field Pressures.
      CALL FLDPRS(FREQ, FLDTYP, NBLKS, IRHSYM, PMATX, VMATX, MDSIZE)

      CALL PRTOUT(FLDTYP, 0)

      DO 220 I = 1, NUMTHP
        WRITE(3,*) THTPHI(1, I) * DEG2RAD, FFP(I)
220   CONTINUE


      DO 230 I = 1, IRECRD(7)
        WRITE(5,*) FREQ, THTPHI(1, I), THTPHI(2, I), PNRMFF(I)
230   CONTINUE

180   CONTINUE
      STOP
      END
```

```
**************************************************************************
CHIEF Driver Program: sc1test1.for
**************************************************************************
C   Program: sc1test1.for
C
C   ***** CONTROL 88 *****

C   PROGRAM CHIEF88 DRIVER
C   MXSREG - MAXIMUM NUMBER OF SURFACE REGIONS
C   MXIPS  - MAXIMUM NUMBER OF INTERIOR POINTS
C   MXARS  - MAXIMUM NUMBER OF SURFACE SUBDIVISIONS/SYM BLK
C   MXGAUS - MAXIMUM ORDER OF GAUSSIAN QUADRATURE
C   MXQPTS - MAXIMUM NUMBER OF QUADRATURE POINTS/SUBDIVISION
C   MXBLKS - MAXIMUM NUMBER OF SYMMETRY BLOCKS
C   MXFFP  - MAXIMUM NUMBER OF FAR-FIELD POINTS
C   MXNFP  - MAXIMUM NUMBER OF NEAR-FIELD POINTS
C   MXPTSC - MAXIMUM NUMBER OF POINT SOURCES
C   MAXCOR - MAXIMUM NUMBER OF FINITE ELEMENT NODES
C   MXFPS - MAXO(MXARS+MXIPS,MXFFP,MXNFP)

    PARAMETER (MXSREG=1200)
    PARAMETER (MAXCOR=5000)
    PARAMETER (MXIPS=20)
    PARAMETER (MXARS=1200)
    PARAMETER (MXGAUS=64)
    PARAMETER (MXQPTS=512)
    PARAMETER (MXBLKS=100)
    PARAMETER (MXFFP=361)
    PARAMETER (MXNFP=361)
    PARAMETER (MXPTSC=20)
    PARAMETER (MXFPS=1250)
C   PARAMETER (MXFPS=MAX0(MXARS+MXIPS,MXFFP,MXNFP))
C   PARAMETER (NWDVEC=2*MXARS)
    PARAMETER (NBLKS = 8)
C
C Input commons
C
C Thin-body and mixed-body commons.  (T. W. Wu)
C
    COMMON /THIN/ ITHIN,IFLAT,NDTHIN,NDEXPD
    COMMON /NORMAL/ PN(4,MXFPS),PNN(4,MXFPS)
    COMMON /MIX/ IMIX,BODY(MXSREG),IBODY(MXARS)
    COMMON /MIX2/ NDSUMP,NDSUMV,NRE,NTH
    CHARACTER*1 BODY
    COMMON /CONST/  RHO,C
    COMMON /EPSLON/ ALT(3,3,3)
    COMMON /PRTCOM/ NUNPRT,NUNERR
    COMMON /PRTRD/  RUNID,DATE
    CHARACTER*32 RUNID
    CHARACTER*8  DATE
    COMMON /NDASG/  NDQPTS,NDPMXS,NDVMXS,NDDECM,NDVELS,NDSPS,
    1          NDPMXF,NDVMXF,NDPMXN,NDVMXN,NDPSSP,NDEXPR,NDCOMV,
```

```
     1          NDTEMP,NDZRDB,NDPATB,NTMPVEL,NTEMP1,NTEMP2,NTEMP3
     COMMON /SVALS/  NSREG,NSEQNS(MXSREG),SUL(MXSREG),SUU(MXSREG),
     1          SVL(MXSREG),SVU(MXSREG),NSU(MXSREG),NSV(MXSREG),
     1          CCS(10,MXSREG),TRNSS(3,MXSREG),IZAX(MXSREG),
     1          IORDU(MXSREG),IORDV(MXSREG),NCCEQS
     COMMON /CORD/  COORDS(MAXCOR,3)
     COMMON /IPTS/  NUMIPS,IPXS(3,MXIPS)
     REAL IPXS
     COMMON /PTSINP/ NUMPTS,PTSRCS(4,MXPTSC),PTSWT(MXPTSC),
     1          IOPTSC(MXPTSC)
     COMPLEX PTSWT
     COMMON /PLWINP/ AINC,THTINC,PHIINC,ISCATR
     COMMON /BAFFLE/ INFBAF
     COMMON /FFINP/  NUMTHP,THTPHI(2,MXFFP)
     COMMON /NFINP/  NUMFPN,NFPXS(3,MXNFP)
     REAL NFPXS
C
C Output commons
C
     COMMON /TAPREC/ RECRD(10),IRECRD(30)
     COMMON /TAPRC1/ ARECRD(10)
     CHARACTER*8 ARECRD
     COMMON /PRGVLS/ NDIMPV,NUMARS,NUMSFP,NUMFFP,NUMNFP,NWDVEC
     COMMON /SURARS/ AREAS(MXARS)
     COMMON /ODSVEC/ TVECT(MXARS),B(MXARS),IPIVTR(MXFPS)
     COMPLEX TVECT, B
     COMMON /VELSPS/ VEL(MXARS),SP(MXARS)
     COMPLEX VEL,SP
     COMMON /PDISL/  POWER,DIRIND,SRCRVL
     COMMON /FFVALS/ FFP(MXFFP),PNRMFF(MXFFP),IFFNRM,RMFNRM
     COMPLEX FFP
     COMMON /TSCOM/  TGTSTH(MXFFP)
     COMMON /NFVALS/ NFP(MXNFP),PNRMNF(MXNFP),INFNRM,RMNNRM
     COMPLEX NFP
     COMMON /PTSCOM/ PTSSP(MXARS)
     COMPLEX PTSSP
     COMMON /EXTCOM/ EXTPRS(MXFPS),IEXTFG
     COMPLEX EXTPRS
     COMMON /NBPRTC/ IRHSPT,NARSPT,NPTBLK,FRQPT
     COMMON /NBPRTS/ SYMTPT
     CHARACTER*3 SYMTPT

C*************** impedance coating modifications ***************
     COMMON /COATING/ ZCOAT, NUMZ
     COMPLEX ZCOAT(MXARS)
C NUMZ - total number of impedance layer surfaces (NSU * NSV)
C*************************************************************

     DIMENSION CC(10), TRNS(3), IELTS(8, 300)
     REAL X1(1000), Y1(1000)

     CHARACTER*3 SYMTYP
```

```
      CHARACTER*4 FLDTYP, TAPEID, PRTTYP
      INTEGER    XIZAX, XNSEQNS, XIRG, XNSU, XNSV, XIORDU, XIORDV
      INTEGER    ICOOR, IELEM
      REAL       THETAA(MXFFP)
      COMPLEX    FFPP(MXFFP)
      COMPLEX   PMATX(148), VMATX(148)
      COMPLEX   ZMTX(4, 4)
      DIMENSION ZRI(2, 4, 4)
      EQUIVALENCE (ZMTX(1,1), ZRI(1,1,1))

      MDSIZE = 148
      RUNID  = 'test1'
      DATE   = '09/07/95'

      CALL INITCM
      CALL OPNSFL

      RHO    = 1000.00
      C      = 1500.00
      NUMZ   = 0
      NUMARS = 4
      OPEN(UNIT=NUNPRT,FILE='test1.out',STATUS='UNKNOWN',
     1    FORM='FORMATTED')


C Symmetry Inputs

      PI = ACOS(-1.0)

      SYMTYP  = 'ROT'
      IRHSYM  = 1
C Surface Region Inputs.

      ROTLIM = PI / NBLKS
      DEG2RAD = PI / 180.0
      XIRG   = 0

      DO 10 I = 1, 10
        CC(I) = 0.0
10    CONTINUE

      DO 20 I = 1, 3
        TRNS(I) = 0.0
20    CONTINUE

C Define region:  rod surfaces (Linear or Quadradic Axisymmetric Interpolation)

      XNSEQNS = 11

C  Read in coordinate and element data.

      ICOOR = 0
```

```
      OPEN(UNIT=8,FILE='iomagc.test1',STATUS='OLD',FORM='FORMATTED')
30    READ(8,*,ERR=50) IDUM, (COORDS(IDUM,J),J=1,2)
      IF (IDUM .GT. ICOOR) ICOOR = IDUM
      IF (IDUM .GT. MAXCOR) THEN
        WRITE (6,*) 'Error:  Coordinate index in coordinate ' //
     +          'data is greater than MAXCOR.'
        STOP
      END IF
      GOTO 30
50    CLOSE(8)

C Read in element data form external file.

      IELEM = 0
      OPEN(UNIT=8,FILE='ioelms.test1',STATUS='OLD',FORM='FORMATTED')
60    READ(8,*,ERR=80) IDUM, (IELTS(J, IELEM+1), J=1,3)
      IELEM = IELEM + 1
      GOTO 60
80    CLOSE(8)

C Display coordinate data.

      WRITE(NUNPRT,90)
90    FORMAT(1H1,'COORDINATE DATA',/)
      DO 110 I = 1, ICOOR
        WRITE(NUNPRT,100) I, (COORDS(I,J), J=1, 2)
100     FORMAT(I5,5X,2E15.4)
110   CONTINUE

C Display element data.

      WRITE(NUNPRT,120)
120   FORMAT(//,' ELEMENT DATA',/)
      DO 140 I = 1, IELEM
        WRITE(NUNPRT,130) I, (IELTS(J,I),J=1,8)
130     FORMAT(I5,10X,8I5)
140   CONTINUE

      TRNS(1) = 0.00000
      TRNS(2) = 0.00000
      TRNS(3) = 0.00000
      XIZAX  = 3
      XSUL   = -1.0
      XSUU   =  1.0
      XSVL   = -ROTLIM
      XSVU   =  ROTLIM
      XNSU   = 1
      XNSV   = 1
      XIORDU = 8
      XIORDV = 8

      DO 150 I = 1, IELEM
```

```
      DO 160 J = 1, 3
        CC(J) = IELTS(J,I)
160     CONTINUE
      XIRG = XIRG + 1
      CALL LDSURR(XIRG, XNSEQNS, CC, TRNS, XIZAX,
     1          XSUL, XSUU, XSVL, XSVU, XNSU, XNSV,
     1          XIORDU, XIORDV)
150   CONTINUE


    NSREG = XIRG

C The syntax for the PLOTCHIEF call is as follows:
C
C CALL PLOTCHIEF(CHARACTER*(*) RUNID,   ! title of CHIEF run.
C           INTEGER     NBLKS,   ! number of symmetry blocks used.
C           CHARACTER*3  SYMTYP,  ! symmetry type from CHIEF.
C           INTEGER     ISUBDIV,  ! 0 - uses optimum subdivisions.
C                       ! 1 - uses NSU's and NSV's defined in LDSURR.
C           REAL        AX,AY,AZ, ! rotation in degrees about the
C                       ! X, Y, and Z axes.
C
C For example:
C    CALL PLOTCHIEF(RUNID, NBLKS, SYMTYP, 1, 30.0, 60.0, 0.0, 0.0, 0.0)

    CALL PRTOUT('GEOM', 1)

    OPEN(UNIT=2,FILE='test1.zrad',STATUS='UNKNOWN')
    OPEN(UNIT=3,FILE='test1.ffpa',STATUS='UNKNOWN')
    OPEN(UNIT=4,FILE='test1.pinc',STATUS='UNKNOWN')
    OPEN(UNIT=30,FILE='test1.tgtr',STATUS='UNKNOWN')

C Define Frequency Loop.

    DO 180 FREQ = 14511.0, 14516.0, 1.00000

C Generate surface P and V matrices.

    CALL SURMAT(FREQ, SYMTYP, NBLKS, PMATX, VMATX, MDSIZE)

C Decompose Matrices

    CALL DECOMM(SYMTYP, NBLKS, IRHSYM, PMATX, VMATX, MDSIZE)

    ISCATR  = 1
    AINC    = 1.00000
    THETAINC = 0.00000
    PHIINC  = 0.00000

    CALL PLWAVE(FREQ, SYMTYP, NBLKS, IRHSYM, AINC, THETAINC, PHIINC)

C Generate Surface Pressures.
```

```
      CALL SURPRS(FREQ, SYMTYP, NBLKS, IRHSYM, PMATX, VMATX, MDSIZE)

      DO 190 i = 1, NUMARS - NUMZ
        WRITE(4,*) REAL(SP(I)) * AREAS(I) * NBLKS/2./PI,
     +      AIMAG(SP(i)) * AREAS(I) * NBLKS/2./PI
  190 CONTINUE
      NUMTHP = 37
      DO 200 I = 1, NUMTHP
        THTPHI(1, I) = (I - 1) * 10 + (0)
        THTPHI(2, I) = 0.0
  200 CONTINUE

      CALL PRTOUT('SP', 1)

      FLDTYP = 'FAR'

C Calculate Far-Field Matrices.
      CALL FLDMAT(FREQ, SYMTYP, FLDTYP, NBLKS, PMATX, VMATX, MDSIZE)

      IFFNRM = 37

C Calculate Far-Field Pressures.
      CALL FLDPRS(FREQ, FLDTYP, NBLKS, IRHSYM, PMATX, VMATX, MDSIZE)

      CALL PRTOUT(FLDTYP, 0)

      DO 210 I = 1, NUMTHP
        WRITE(3,*) THTPHI(1, I) * DEG2RAD, FFP(I)
  210 CONTINUE

      DO 220 I = 1, NUMTHP
        WRITE(30,*) THTPHI(1, I), TGTSTH(I)
  220 CONTINUE

C Generate radiation impedance matricies.

      NUMZ = 0
      CALL ZRADMX(FREQ, ZMTX, NUMARS, SYMTYP, NBLKS,
     1        PMATX, VMATX, MDSIZE)

      NUMZ = 0
      WRITE(2,*) ((ZRI(1,I,J)*NBLKS/2.0/PI, J=1,NUMARS-NUMZ),
     &            I=1,NUMARS-NUMZ)
      WRITE(2,*) ((ZRI(2,I,J)*NBLKS/2.0/PI, J=1,NUMARS-NUMZ),
     &            I=1,NUMARS-NUMZ)
  180 CONTINUE
      STOP
      END
```

```
************************************************************************
CHIEF Driver Program: sc2test1.for
************************************************************************
C   Program: sc2test1.for
C
C   ***** CONTROL 88 *****

C   PROGRAM CHIEF88 DRIVER
C   MXSREG - MAXIMUM NUMBER OF SURFACE REGIONS
C   MXIPS  - MAXIMUM NUMBER OF INTERIOR POINTS
C   MXARS  - MAXIMUM NUMBER OF SURFACE SUBDIVISIONS/SYM BLK
C   MXGAUS - MAXIMUM ORDER OF GAUSSIAN QUADRATURE
C   MXQPTS - MAXIMUM NUMBER OF QUADRATURE POINTS/SUBDIVISION
C   MXBLKS - MAXIMUM NUMBER OF SYMMETRY BLOCKS
C   MXFFP  - MAXIMUM NUMBER OF FAR-FIELD POINTS
C   MXNFP  - MAXIMUM NUMBER OF NEAR-FIELD POINTS
C   MXPTSC - MAXIMUM NUMBER OF POINT SOURCES
C   MAXCOR - MAXIMUM NUMBER OF FINITE ELEMENT NODES
C   MXFPS - MAXO(MXARS+MXIPS,MXFFP,MXNFP)

    PARAMETER (MXSREG=1200)
    PARAMETER (MAXCOR=5000)
    PARAMETER (MXIPS=20)
    PARAMETER (MXARS=1200)
    PARAMETER (MXGAUS=64)
    PARAMETER (MXQPTS=512)
    PARAMETER (MXBLKS=100)
    PARAMETER (MXFFP=361)
    PARAMETER (MXNFP=361)
    PARAMETER (MXPTSC=20)
    PARAMETER (MXFPS=1250)
C    PARAMETER (MXFPS=MAX0(MXARS+MXIPS,MXFFP,MXNFP))
C    PARAMETER (NWDVEC=2*MXARS)
    PARAMETER (NBLKS = 8)
C
C Input commons
C
C Thin-body and mixed-body commons.  (T. W. Wu)
C
    COMMON /THIN/ ITHIN,IFLAT,NDTHIN,NDEXPD
    COMMON /NORMAL/ PN(4,MXFPS),PNN(4,MXFPS)
    COMMON /MIX/ IMIX,BODY(MXSREG),IBODY(MXARS)
    COMMON /MIX2/ NDSUMP,NDSUMV,NRE,NTH
    CHARACTER*1 BODY
    COMMON /CONST/  RHO,C
    COMMON /EPSLON/ ALT(3,3,3)
    COMMON /PRTCOM/ NUNPRT,NUNERR
    COMMON /PRTRD/  RUNID,DATE
    CHARACTER*32 RUNID
    CHARACTER*8  DATE
    COMMON /NDASG/  NDQPTS,NDPMXS,NDVMXS,NDDECM,NDVELS,NDSPS,
   1          NDPMXF,NDVMXF,NDPMXN,NDVMXN,NDPSSP,NDEXPR,NDCOMV,
```

```
     1          NDTEMP,NDZRDB,NDPATB,NTMPVEL,NTEMP1,NTEMP2,NTEMP3
      COMMON /SVALS/  NSREG,NSEQNS(MXSREG),SUL(MXSREG),SUU(MXSREG),
     1          SVL(MXSREG),SVU(MXSREG),NSU(MXSREG),NSV(MXSREG),
     1          CCS(10,MXSREG),TRNSS(3,MXSREG),IZAX(MXSREG),
     1          IORDU(MXSREG),IORDV(MXSREG),NCCEQS
      COMMON /CORD/   COORDS(MAXCOR,3)
      COMMON /IPTS/   NUMIPS,IPXS(3,MXIPS)
      REAL IPXS
      COMMON /PTSINP/ NUMPTS,PTSRCS(4,MXPTSC),PTSWT(MXPTSC),
     1          IOPTSC(MXPTSC)
      COMPLEX PTSWT
      COMMON /PLWINP/ AINC,THTINC,PHIINC,ISCATR
      COMMON /BAFFLE/ INFBAF
      COMMON /FFINP/  NUMTHP,THTPHI(2,MXFFP)
      COMMON /NFINP/  NUMFPN,NFPXS(3,MXNFP)
      REAL NFPXS
C
C Output commons
C
      COMMON /TAPREC/ RECRD(10),IRECRD(30)
      COMMON /TAPRC1/ ARECRD(10)
      CHARACTER*8 ARECRD
      COMMON /PRGVLS/ NDIMPV,NUMARS,NUMSFP,NUMFFP,NUMNFP,NWDVEC
      COMMON /SURARS/ AREAS(MXARS)
      COMMON /ODSVEC/ TVECT(MXARS),B(MXARS),IPIVTR(MXFPS)
      COMPLEX TVECT, B
      COMMON /VELSPS/ VEL(MXARS),SP(MXARS)
      COMPLEX VEL,SP
      COMMON /PDISL/  POWER,DIRIND,SRCRVL
      COMMON /FFVALS/ FFP(MXFFP),PNRMFF(MXFFP),IFFNRM,RMFNRM
      COMPLEX FFP
      COMMON /TSCOM/  TGTSTH(MXFFP)
      COMMON /NFVALS/ NFP(MXNFP),PNRMNF(MXNFP),INFNRM,RMNNRM
      COMPLEX NFP
      COMMON /PTSCOM/ PTSSP(MXARS)
      COMPLEX PTSSP
      COMMON /EXTCOM/ EXTPRS(MXFPS),IEXTFG
      COMPLEX EXTPRS
      COMMON /NBPRTC/ IRHSPT,NARSPT,NPTBLK,FRQPT
      COMMON /NBPRTS/ SYMTPT
      CHARACTER*3 SYMTPT

C*************** impedance coating modifications ***************
      COMMON /COATING/ ZCOAT, NUMZ
      COMPLEX ZCOAT(MXARS)
C  NUMZ - total number of impedance layer surfaces (NSU * NSV)
C*************************************************************

      DIMENSION CC(10), TRNS(3), IELTS(8, 300)
      REAL X1(1000), Y1(1000)

      CHARACTER*3 SYMTYP
```

```
      CHARACTER*4 FLDTYP, TAPEID, PRTTYP
      INTEGER    XIZAX, XNSEQNS, XIRG, XNSU, XNSV, XIORDU, XIORDV
      INTEGER    ICOOR, IELEM
      REAL       THETAA(MXFFP)
      COMPLEX    FFPP(MXFFP)
      COMPLEX   PMATX(148), VMATX(148)
      MDSIZE = 148
      RUNID  = 'test1'
      DATE   = '09/07/95'

      CALL INITCM
      CALL OPNSFL

      RHO    = 1000.00
      C      = 1500.00
      NUMZ   = 0
      NUMARS = 4
      OPEN(UNIT=NUNPRT,FILE='test1.out',STATUS='UNKNOWN',
     1    FORM='FORMATTED')


C Symmetry Inputs

      PI = ACOS(-1.0)

      SYMTYP  = 'ROT'
      IRHSYM  = 1
C Surface Region Inputs.

      ROTLIM = PI / NBLKS
      DEG2RAD = PI / 180.0
      XIRG   = 0

      DO 10 I = 1, 10
        CC(I) = 0.0
10    CONTINUE

      DO 20 I = 1, 3
        TRNS(I) = 0.0
20    CONTINUE

C Define region:  rod surfaces (Linear or Quadradic Axisymmetric Interpolation)

      XNSEQNS = 11

C Read in coordinate and element data.

      ICOOR = 0
      OPEN(UNIT=8,FILE='iomagc.test1',STATUS='OLD',FORM='FORMATTED')
30    READ(8,*,ERR=50) IDUM, (COORDS(IDUM,J),J=1,2)
      IF (IDUM .GT. ICOOR) ICOOR = IDUM
      IF (IDUM .GT. MAXCOR) THEN
```

```
      WRITE (6,*) 'Error:  Coordinate index in coordinate ' //
     +           'data is greater than MAXCOR.'
        STOP
      END IF
      GOTO 30
50    CLOSE(8)

C Read in element data form external file.

      IELEM = 0
      OPEN(UNIT=8,FILE='ioelms.test1',STATUS='OLD',FORM='FORMATTED')
60    READ(8,*,ERR=80) IDUM, (IELTS(J, IELEM+1), J=1,3)
      IELEM = IELEM + 1
      GOTO 60
80    CLOSE(8)

C Display coordinate data.

      WRITE(NUNPRT,90)
90    FORMAT(1H1,'COORDINATE DATA',/)
      DO 110 I = 1, ICOOR
        WRITE(NUNPRT,100) I, (COORDS(I,J), J=1, 2)
100     FORMAT(I5,5X,2E15.4)
110   CONTINUE

C Display element data.

      WRITE(NUNPRT,120)
120   FORMAT(//,' ELEMENT DATA',/)
      DO 140 I = 1, IELEM
        WRITE(NUNPRT,130) I, (IELTS(J,I),J=1,8)
130     FORMAT(I5,10X,8I5)
140   CONTINUE

      TRNS(1) = 0.00000
      TRNS(2) = 0.00000
      TRNS(3) = 0.00000
      XIZAX  = 3
      XSUL   = -1.0
      XSUU   =  1.0
      XSVL   = -ROTLIM
      XSVU   =  ROTLIM
      XNSU   = 1
      XNSV   = 1
      XIORDU = 8
      XIORDV = 8

      DO 150 I = 1, IELEM
        DO 160 J = 1, 3
          CC(J) = IELTS(J,I)
160     CONTINUE
        XIRG = XIRG + 1
```

```
      CALL LDSURR(XIRG, XNSEQNS, CC, TRNS, XIZAX,
     1          XSUL, XSUU, XSVL, XSVU, XNSU, XNSV,
     1          XIORDU, XIORDV)
  150 CONTINUE


      NSREG = XIRG

C The syntax for the PLOTCHIEF call is as follows:
C
C CALL PLOTCHIEF(CHARACTER*(*) RUNID,   ! title of CHIEF run.
C           INTEGER     NBLKS,   ! number of symmetry blocks used.
C           CHARACTER*3  SYMTYP,  ! symmetry type from CHIEF.
C           INTEGER     ISUBDIV,  ! 0 - uses optimum subdivisions.
C                       ! 1 - uses NSU's and NSV's defined in LDSURR.
C           REAL        AX,AY,AZ, ! rotation in degrees about the
C                       ! X, Y, and Z axes.
C
C For example:
C     CALL PLOTCHIEF(RUNID, NBLKS, SYMTYP, 1, 30.0, 60.0, 0.0, 0.0, 0.0)

      CALL PRTOUT('GEOM', 1)

      OPEN(UNIT=2,FILE='test1.ffpa',STATUS='OLD')
      OPEN(UNIT=3,FILE='test1.ffpb',STATUS='UNKNOWN')
      OPEN(UNIT=30,FILE='test1.tgtt',STATUS='UNKNOWN')
      OPEN(UNIT=4,FILE='test1.vel',STATUS='OLD')

C Define Frequency Loop.

      DO 180 FREQ = 14511.0, 14516.0, 1.00000

C Generate surface P and V matrices.

      CALL SURMAT(FREQ, SYMTYP, NBLKS, PMATX, VMATX, MDSIZE)

C Decompose Matrices

      CALL DECOMM(SYMTYP, NBLKS, IRHSYM, PMATX, VMATX, MDSIZE)

      CALL IOSUB(NDVELS, 10, VEL, 0)

      DO 190 I = 1, 4
        READ(4,*) VEL(I)
  190 CONTINUE

      CALL IOSUB(NDVELS, 1, VEL, NWDVEC)

C Generate Surface Pressures.

      CALL SURPRS(FREQ, SYMTYP, NBLKS, IRHSYM, PMATX, VMATX, MDSIZE)
```

```
      CALL PRTOUT('VEL', 1)

      CALL PRTOUT('SP', 1)

      FLDTYP = 'FAR'

      NUMTHP = 703
      ICOUNT = 0
      DO 210 I = 0, 180, 10
        DO 200 J = 0, 360, 10
          ICOUNT = ICOUNT + 1
          THTPHI(1,ICOUNT) = J
          THTPHI(2,ICOUNT) = I
200     CONTINUE
210   CONTINUE

C  Calculate Far-Field Matrices.
      CALL FLDMAT(FREQ, SYMTYP, FLDTYP, NBLKS, PMATX, VMATX, MDSIZE)

      IFFNRM = 19

C  Calculate Far-Field Pressures.
      CALL FLDPRS(FREQ, FLDTYP, NBLKS, IRHSYM, PMATX, VMATX, MDSIZE)

      CALL PRTOUT(FLDTYP, 0)

      DO 220 I = 1, NUMTHP
        READ(2,*) THETAA(I), FFPP(I)
        WRITE(3,*) THTPHI(1, I) * DEG2RAD, FFP(I)
        WRITE(30,*) THTPHI(1, I) * DEG2RAD,
     &          20. * ALOG10(CABS(FFP(I) + FFPP(I))/AINC)
220   CONTINUE

      CALL PRTOUT('VEL', 1)

180   CONTINUE
      STOP
      END
```

REFERENCES

1. O. C. Zienkiewicz, *The Finite Element Method, 3rd Edition,* McGraw-Hill, London, 1977.

2. H. Schenck, "Improved Integral Formulation for Acoustic Radiation Problems," *Journal of the Acoustical Society of America,* vol. 44, 1968, pp. 41-58.

3. P. Bettess, *Infinite Elements*, Penshaw, Sunderland, England, 1992.

4. D. T. Wilton, "Acoustic Radiation and Scattering from Elastic Structures," *International Journal for Numerical Methods in Engineering,* vol. 13, 1978, pp. 123-138.

5. H. A. Schenck and G. W. Benthien, "Numerical Solution of Acoustic-Structure Interaction Problems," NOSC Technical Report 1263, Naval Oceans Systems Center, San Diego, CA, April 1989.

6. C. Siders, P. Klippel, and D. Bollings, "An Introduction to XCID Including Recent Enhancement to CHIEF," NUWC-NPT Technical Document 10,518, Naval Undersea Warfare Center Division, Newport, RI, 30 November 1995.

7. P. K. Klippel, Naval Undersea Warfare Center Division, Newport, RI 02841-1708.

8. O. B. Wilson, *Introduction to Theory and Design of Sonar Transducers*, Peninsula Publishing, Los Altos, CA, 1988, ch. 4.

9. O. B. Wilson, *Introduction to Theory and Design of Sonar Transducers*, Peninsula Publishing, Los Altos, CA, 1988, ch. 3.

10. P. Businger and G. H. Golub, "Linear Least Squares Solutions by Householder Transformations," *Numerische Mathematik, vol. 7,* 1965, pp. 269-279.

11. M. C. Junger and D. Feit, *Sound, Structures, and Their Interaction*, MIT Press, Cambridge, MA, 1972, pp. 315-320.

12. J. N. Decarpigny and J. C. Debus, "In-Air Analysis of Piezoelectric Tonpilz Transducers in a Wide Frequency Band Using a Mixed Finite Element-Plane Wave Method," *Journal of the Acoustical Society of America,* vol. 78, 1985, pp. 1499-1507.

13. G. Benthien, "Application of Frequency Interpolation to Acoustic-Structure Interaction Problems," NOSC Technical Report 1323, Naval Ocean Systems Center, San Diego, CA November 1989.

14. J. P. Coyette and K. R. Fyfe, "Solution of Elasto-Acoustic Problems Using a Variational Finite Element/Boundary Element Technique," *Numerical Techniques in Acoustic Radiation*, R. J. Bernhard and R. F. Keltie, eds., American Society of Mechanical Engineers, New York, 1989, NCA-vol. 6, pp. 15-25.

15. M. D. McCollum and C. M. Siders, "Modal Analysis of a Structure in a Compressible Fluid Using a Finite Element/Boundary Element Approach," (accepted for publication by the *Journal of the Acoustical Society of America*).

16. G. Benthien, "Contributions of Individual Structural Modes to the Scattered Acoustic Field," NOSC Technical Report 1329, Naval Ocean Systems Center, San Diego, CA, November 1989.

17. A. L. Van Buren, "A Test of the Capabilities of CHIEF in the Numerical Calculation of Acoustic Radiation from Arbitrary Surfaces," NRL Report 7160, Naval Research Laboratory, Washington, DC, November 1970.

18. O. C. Zienkiewicz, *The Finite Element Method in Engineering Science*, McGraw-Hill, London, 1971, p. 81.

19. E. Anderson, Z. Bai, C. Bischof, J. Demmel, J. Dongarra, J. Du Croz, A. Greenbaum, S. Hammarling, A. McKenney, S. Ostrouchov, and D. Sorensen, LAPACK Users' Guide, Society for Industrial and Applied Mathematics, Philadelphia, PA, 1992.

20. D. A. Berlincourt, D. R. Curran, and H. Jaffe, *Physical Acoustics, Vol. I, Part A*, W. P. Mason, ed., Academic Press, New York, 1964, p. 227.

21. B. A. Auld, *Acoustic Fields and Waves in Solids, Vol. I*, John Wiley and Sons, New York, 1973, ch. 6.

22. A. I. Beltzer, *Acoustics of Solids*, Springer-Verlag, Berlin, 1988, ch. 3.

23. E. Dieulesaint and D. Royer, *Elastic Waves in Solids*, John Wiley and Sons, New York, 1980, ch. 6.

24. H. Allik and T. J. R. Hughes, "Finite Element Method for Piezoelectric Vibration," *International Journal of Numerical Methods in Engineering*, vol. 2, 1970, pp. 151-157.

25. R. R. Smith, J. T. Hunt, and D. Barach, "Finite Element Analysis of Acoustically Radiating Structures with Applications to Sonar Transducers, *Journal of the Acoustical Society of America*, vol. 54, 1973, pp. 1277-1288.

26. H. Allik, K. M. Webman, and J. T. Hunt, "Vibration Response of Sonar Transducers Using Piezoelectric Finite Elements," *Journal of the Acoustical Society of America*, vol. 56, no. 6, 1974, pp. 1782-1791.

27. J. T. Hunt, M. R. Knittel, and D. Barach, "A Finite Element Approach to Acoustic Radiation from Elastic Structures," *Journal of the Acoustical Society of America*, vol. 55 , 1974, pp. 269-280.

28. Private communication with A. F. Seybert, Department of Mechanical Engineering, University of Kentucky, Lexington, KY, April 1990.

INITIAL DISTRIBUTION LIST

Addressee	No. of Copies
Office of Naval Research (Code 312SS -- S. Littlefield; Code 331 -- W. Smith, R. Vogelsong; Code 334 -- G. Main, A. Tucker)	5
NASA, Langley, VA (M/S 166 -- M. Brown)	1
Naval Research Laboratory (Code 7100; Code 7130 -- J. Bucaro; Code 5130 -- K. Washburn)	3
Naval Postgraduate School (Code 61W1 -- S. Baker, O. Wilson; Technical Library)	3
Naval Air Systems Command (Technical Library)	1
Space and Naval Warfare Systems Command (PMW-182)	1
Naval Sea Systems Command (Technical Library)	1
Naval Undersea Warfare Center, Keyport Division	1
National Oceanic and Atmospheric Administration (Ocean Acoustics -- J. Stamates)	1
Naval Surface Warfare Center, Carderock Division (Code 1925 -- D. Heydrick; Code 2711 -- R. McConnell; Code 5222 -- Library)	3
Naval Surface Warfare Center, Dahlgren Division (Technical Library, Acoustics Division)	2
Naval Surface Warfare Center, Coastal Systems Station (Technical Library)	1
Naval Surface Warfare Center, Crane Division, (Library)	1
Naval Command Control and Surveillance Center, RDT&E Division, San Diego (Code 712 -- S. Nichols; Code 9642 -- E. Rynne, G. Benthien, D. Gillette, Technical Library)	5
Naval Air Warfare Center (Technical Library)	1
Center for Naval Analyses	1
Defense Technical Information Center	12
Applied Research Laboratory, Pennsylvania State University (W. Thompson; Bioengineering -- W. Cao)	2
Boeing Defense and Space Group, Huntsville, AL (E. O'Keefe)	1
Engineering Acoustics, Winter Park, FL (T. Ensign)	1
Hazeltine, Braintree, MA (D. Merrell)	1
Hydroacoustics, Inc, Rochester, NY (R. DeLaCroix)	1
Image Acoustics, Cohasset, MA (J. Butler)	1
International Transducer Corporation, Santa Barbara, CA (E. Kunstal)	1
Lockheed Missile and Space Division, Cupertino, CA (B. Thorn)	1
MAR, Inc, Ft. Lauderdale, FL (B. Willard)	1
MITRE Corporation, McLean, VA (C. Burmaster)	1
Raytheon Company, Portsmouth, RI (M. Ring)	1
RD Instruments, San Diego, CA (R. Merewether)	1
Sonatech, Inc, Santa Barbara, CA (R. Grimes-Graeme)	1
Sparton Electronics, Jackson, MI (T. Pawela)	1
Sparton Defense Electronics, DeLeon Springs, FL (G. Martin)	1
Technical Consultation Services, Columbus, OH (R. Simon)	1
Texas Research International, Austin, TX (J. Thornton)	1
Westinghouse Electric Corporation, Clevelend, OH (M. Johnson, S. Thompson)	2
Westlake Research Applications, Westlake Village, CA (C. Yu)	1
California State University Long Beach (S. McInerny)	1
Florida Atlantic University (J. Cuschieri)	1
Georgia Institute of Technology (A. Ferri)	1
Stanford University (S. Williams)	1
University of Kentucky (A. Seybert, T. Wu)	2
Celestion International Ltd, Ipswich, England (J. Wright)	1
Imperial College, London (I. Mathews)	1
Stanley Ehrlich, Newport, RI	1
David Carson, San Diego, CA	1
Mark Hayner, Reading, MA	1